2012年1月12日発行
第7巻第4号（通巻29号）
ISSN 1349-8622

巻頭言
播磨 崇 特定非営利活動法人 ITコーディネータ協会 会長

所長対談：野辺 継男 日産自動車株式会社 ビークルインフォメーションテクノロジー事業本部 General Manager

「外部と繋がる車」がもたらす未来とITが果たす役割を考える

連載 情報システムの障害データ

情報システムの障害状況 2011年前半データ

SEC journal論文賞 受賞論文発表

IPA FORUM 2011 招待講演より

ソフトウェアの品質保証とテスト

～メトリクスと測定、明確な記述、そして管理可能なプロセスの三本の柱が
より信頼できるソフトウェア開発に寄与する～
ポール・イー・ブラック博士

技術解説
ソフトウェアの品質説明力強化の取り組み
消費者機械安全性・信頼性保証の国際標準化

アングル
形式手法導入のための産学連携PBLの活用

組織紹介
一般社団法人TERASの紹介（前編）
CEA-LIST フランス原子力・代替エネルギー庁システム統合技術応用研究所

Column
就職はなぜ難しいのか

独立行政法人情報処理推進機構
http://www.ipa.go.jp/

SEC journal No. 27
独立行政法人情報処理推進機構

SEC journal No.27
2012年1月12日発行
第7巻第4号（通巻29号）
ISSN 1349-8622

巻頭言

中小企業のIT経営を支える人財～ITコーディネータ～

特定非営利活動法人
ITコーディネータ協会　会長
播磨　崇

ITコーディネータ協会は今年で発足以来11年目に入った。東日本大震災など社会情勢が厳しい中、これからが本当の正念場である。現在、協会は、多方面にわたり改革の取り組みを始めている。そのような重要な時期である2011年6月、新たに会長に就任したことに身の引き締まる思いである。

中小企業のIT化を支えるITコーディネータ

ITコーディネータ制度は、中小企業のIT化を促進し、日本の中小企業の競争力を高めるということを狙いにして（経済産業省の肝いりで）2001年に発足した。現在までに、6,700名のITコーディネータを輩出しており、全国に200強のITコーディネータの組織（届出組織）がある。届出組織は地域ごとにITコーディネータを組織化し、中小企業の支援の機動力を高めている。

ITコーディネータ協会の主な活動としては、全国のITコーディネータがより高い満足度で中小企業の経営者に対応出来るよう以下の3点を中心に、種々の活動を行っている。

・ITコーディネータのビジネスの活性化
・ITコーディネータ資格者の認定・育成施策
・ITコーディネータが活用する技術面の整備

これからの中小企業のITの利活用のあり方

中小企業のIT化の実情を見てみると、投資余力、保有人財、IT化スキル面から見て、大きな制約を抱えている。新しいコンピュータの利活用のスタイルであるクラウドコンピューティングは一つの解決策を提供するものとして期待されている。しかし、提供側であるITベンダーも、ビジネスが緒に就いたところであり、中小企業の期待に機能面、価格面で、十分な満足を与えるところまでいっていない。ITコーディネータ協会は、商工団体、ITベンダーと手を組み、「中小企業支援SaaS利用促進コンソーシアム」を立ち上げ、これらの課題の解決に取り組んでいる。

クラウド環境でのビジネス推進においては、セキュリティ、契約、SLA等の整備が一層重要になってくる。クラウド化ビジネスの技術基盤の整備は急務であり、今後ともIPAとの連携が必要である。

中小ITベンダーの業態変革に向け

大多数の中小企業にはこれまで、中堅・中小ITベンダー、IT販社がかかわってきているが、今後の開発中心需要の減少に対応した多重下請け構造からの脱却並びにユーザの経営改善ニーズ（顧客創造・商品／サービス付加価値向上・収益拡大、新たなビジネスモデル作り）に応える上流ビジネスへの付加価値シフト等、今大きな業態変革が問われている。

ITコーディネータ協会として、IPAとも連携を取り、これらの問題に自己完結型の対応をするのではなく、より広く知見を求めていく必要性を痛感している。IPA全体の取り組み、とりわけIPA/SECの取り組みとうまく連携させていただき、難しい局面を打開していきたい。

「外部と繋がる車」がもたらす未来とITが果たす役割を考える

日産自動車株式会社　ビークルインフォメーションテクノロジー事業本部
General Manager
野辺 継男

SEC 所長
松田 晃一

今、自動車が外部と繋がる世界が誕生している。通話だけでなく、様々なサービスが利用可能になりつつある。これら新たな展開における、自動車向けのサービスの行方、そしてITが果たす役割についてお話を伺った。

松田：本日は、「外部と繋がる車」をテーマにして、車の繋がりがどのように広がっていくのか、そしてそこにITやソフトウェアがどのような役割を果たすのかについて考えていきたいと思います。過去、自動車は外部と閉ざされた空間であり、いったん自動車に乗ってしまうと外部の情報が入ってこない環境でした。せいぜいカーラジオから情報が得られる程度だったわけです。それが1979年末に登場した自動車電話から始まって少しずつ車の環境と外部の環境が繋がるようになってきました。

野辺：自動車は移動するので外部と繋げる手段は必然的に無線になります。携帯電話の始まりは自動車電話といえますが、可搬出来る電話とはいえ、電話機自体がまだ大きく重かったので、携帯電話は自動車から始まったという事情もあったと思います。

野辺 継男（のべ つぐお）
1983年早稲田大学理工学部卒。1988～1990年HBS MBA、1989～1990年同大学PIRPフェロー。1983年日本電気入社。パソコンを核に国内外で各種事業立上げ（VOD、データ放送、各種インターネットソリューション）。2000年末退職後オンラインゲーム会社立ち上げ、CEO。2004年日産自動車入社。テレマティクス統括。LEAFとPCやスマートフォンとの接続、3rd Partyソリューションとの連携インフラ構築。

松田：自動車電話は、約7kgの本体をトランクルームに設置してボディ後部にアンテナを立て、運転席の横には送受話器だけをセットするという仕組みでスタートしました。車にアンテナがついていることがステータスの時代でした。その頃から車が外部環境と繋がるようになってきましたね。

野辺：携帯電話でデータ通信が出来るようになったのは1999年のことで、たった10余年で日常的に使用されるようになりました。海外の携帯電話の場合は、音声通話の浸透は同様に早かったのですが、最近までデータ通信契約率は低い状況でした。ところがiPhoneの登場により、米国でも携帯電話でデータ通信をすることが浸透しました。また、エレクトロニクス部品についても、コンシューマ製品にも自動車に搭載出来るほどの信頼性のあるものが徐々に採用されるようになり、量が出て結果的に車載通信機器の低コスト化に寄与しています。

松田：無線データ通信以外に、道路の渋滞情報をカーナビに取り込むVICS（Vehicle Information and Communication System）などITS（Intelligent Transport Systems：高度道路交通システム）分野の技術もありますが、他の通信手段はどのように進展しているのでしょうか。

野辺：ITSもどんどん進化しています。ITSベースの交通情報は路上等に設置されたインフラ設備からデータを取り込んで、多くはFMのアナログ波を使って情報を提供しています。ですから、通信ではなく放送に近い状況です。携帯の実質データ通信速度はおよそ1Mbpsと高速ですが、FM多重放送の場合16kbps。速度が圧倒的に違いますが、FM放送がデジタル化されれば、より広範囲に大量のデータを短時間に送ることが可能になります。それによってVICSから提供する交通情報もリアルタイム性を高めることが出来ると思います。そのほかの通信手段としては、DSRC、Wi-FiやWiMAXもあります。また、車車間通信にはミリ波レーダーや超音波も使われています。

常時接続と安価な定額料金が重要な要件

松田：車が繋がること。それを加速させる要件についてどのように考えられていますか。

野辺：ブロードバンド化というと通信速度の向上が注目されます

が、それ以上に安定した常時接続環境と安い定額料金、これが重要です。家庭やオフィスでも、ADSL や光回線などといった安価な常時接続の普及により、インターネットが広く使われるようになりました。定額制により、料金を気にしなくて済み、常時接続によりリアルタイムでインタラクティブな情報配信が可能になった、という使い勝手の向上が普及に繋がったと思います。自動車でも LTE（Long Term Evolution）になると、使い勝手の向上が見込めます。

松田：車の場合、使い勝手はどう変わりますか？

野辺：繋がりっぱなしになると、プローブベースの渋滞情報（タイムスタンプ付の位置情報をサーバーにアップして統計処理後交通情報とするもの）の提供･取得のリアルタイム性が高くなります。

松田：それは、東日本大震災のときに「通れた道マップ」を Google のマップで見られるようにした技術ですね。

野辺：元々は、各社さんが自社の自動車からアップした情報を自分の会社のために使っていたものですが、東日本大震災のときに、その情報を Google の KML というデータフォーマットに合わせてアップして Google マップで提供したものです。

松田：マスコミでも取り上げられましたが、非常に有効だったのですね。その情報は常時接続で自動車からサーバーに送られたのですか。

野辺：残念ながら現在はまだ常時接続にはなっていません。まとめた情報をある段階で一気にアップしたりダウンロードしています。まず、車に乗ってキーをイグニション・オンにすると通信を行います。そのあとはおおむね通信を止めておく状態になります。ユーザー設定によっては、30分に1回とか5分に1回アップするというようなことも可能です。

松田：カーナビの情報を携帯電話を使って送信するわけですが、今後は、車にデータ通信機能が内蔵されていくのでしょうか。

野辺：既に高級車や電気自動車ではデータ通信機能を内蔵している車種があります。日産の電気自動車でいうと LEAF はデータ通信機能を内蔵しています。これから、何らかのデータ通信機能を内蔵した車が一般的になってくると思います。

松田：データ通信機能を内蔵した車が一般的になるためのカギはやはりコストですね。

野辺：その通りです。そのために量が多くないと安くはなりません。車の国際的な年間売上はだいたい 6,000 ～ 7,000 万台。中国市場が成長しているので5～6年後には合計 8,000 万台になると見られています。それに対してスマートフォンは iOS という一つの OS だけでも1億台の市場があるわけです。5～6年たてばスマートフォン市場は全体で 10 億台を超えるでしょう。すると、放送系やミリ波レーダー系のソリューションは追いつけないほどの低コストになる可能性があります。これから車の中で使われるサービスは、携帯電話を通信手段として使ったものが多くなっていくと思います。

“リレンダリング”機能が車載機器には重要

松田：今は、ドライバーの携帯電話を使い通信をする形になっていますが、車に通信機能が内蔵されると、ドライバーは通信のことを意識する必要がなくなるわけですね。

野辺：おっしゃる通りです。しかし、車にスマートフォンをそのまま車載化するのは困難で、スマートフォンと同様の技術を採用し、スマートフォンと通信しあうインテリジェントなディスプレイユニットが必要になると考えています。ディスプレイに映し出す情報やユーザインターフェースもスマートフォンと異なるものになります。今後は HTML5 がその方向性を加速するでしょう。

松田：なるほど。どのような利用が想定されますか。

野辺：例えば、スマートフォンではレストランの予約が容易に出来ますが、時間の変更やキャンセルをするときには、再度そのレストランを検索し、そのうえで種々の操作を行います。しかし、スマートフォンで予約したのなら、それに接続されている車載のインテリジェントディスプレイユニットには、例えば直近のアクセス履歴から予約レストランボタンを再表示することが可能です。そのボタンを押すだけで、ドライバーが持ち込んだスマートフォンからレストランに電話が繋がり、「遅れます」と連絡出来るようになります。車を運転していて行いたいのは、新たに検索したり予約する行為よりも、既に行った予約の再確認や変更です。つまり、スマートフォンで1回行ったことを再現するレンダリング機能、即ち“リレンダリング”が、車載機には重要です。運転しながら始めての情報を検索するということは実はあまり行われず、既に行った情報アクセスを再確認し、時に修正する。それによってドライバーの運転中の不安要素を排除するわけです。今後、ドライバーがよりいっそう車の運

松田 晃一（まつだ こういち）

1970年京都大学大学院修士課程修了後、日本電信電話公社入社。NTTソフトウェア研究所ソフトウェア開発技術研究部長、株式会社国際電気通信基礎技術研究所（ATR）取締役企画部長、NTTコミュニケーション科学研究所 所長、NTT先端技術総合研究所所長、NTTアドバンステクノロジ株式会社代表取締役常務、NTT AT IPシェアリング株式会社代表取締役社長を歴任し、2008年2月IPA（独立行政法人情報処理推進機構）IT人材育成本部長に就任、2009年1月よりSEC（ソフトウェア・エンジニアリング・センター）所長。工学博士。

転に集中出来るようにするために、ドライバーの思考パターンをシミュレーションし、それを支援するユーザインタフェースやアルゴリズムを考えていく必要があります。

アドバンストサービスプロバイダと連携

松田：自動車が繋がることによって、どのようなサービスが実現されるのでしょうか。

野辺：自動車会社のサーバーをサードパーティのサービスプロバイダに繋ぐことにより、車にサービスプロバイダのサービスを簡単に提供出来るようになります。ガソリンスタンドや電気自動車のチャージングスポット、またスマートグリッドなどに繋げば、どのくらいの車がどの辺りの場所でどのくらいの電気をチャージしようとしているのかという情報を、電気を供給する側に提供することが出来ます。

保険会社のサービス提供にも利用出来ます。ヨーロッパで浸透している保険商品にPay As You Driveというものがあります。乗った距離や運転の仕方に基づいて保険料率を決める仕組みです。ほかにも、ドライバーがコンビニエンスストアに行こうとしていたら、ターゲットマーケティングによりドライバーに合わせた推奨商品を画面に表示する、といったことも技術的には可能です。また、電気自動車に充電中のドライバーの嗜好に合わせて映画の映像をディスプレイに配信してDVDの販売促進をしたり、ロードショーのチケットを割引販売するなど、デジタルサイネージとしてサービスすることもいずれ可能になるでしょう。

松田：車から集めたいろいろな種類の情報を基にして、新しいサービスを提供出来る可能性があるのですね。

野辺：ドライバーのニーズとサービスプロバイダのニーズをマッチングさせることによって、新しい事業機会が生まれるわけです。こうしたサービスを最近ではアドバンストサービスといっています。海外では最近、ITというとシステムインテグレーションを指します。そのため、例えば、GoogleはIT企業とは呼ばれず、アドバンストサービスプロバイダ等と表されています。車についてもアドバンストサービスが重要になるという表現をしています。

松田：日産自動車としてはどのようなサービスをしているのですか。

野辺：日産はLEAFに内蔵電話を搭載しました。この内蔵電話は日産のグローバルデータセンター（CARWINGSデータセンター）と繋がっています。それによってドライバーにサービスを提供しています。例えば、Plug-in Reminderがあります。これは、LEAFの充電を、位置情報を使ってサポートする機能で、もし帰宅後に充電のためのプラグを挿し忘れていたら、あるタイミングで充電を促すメールを送信するというサービスです。

このように、車とデータセンターが、つながることの重要な点は、充電が必要なことを知らせるにとどまらず、日産のサーバーと外部のサーバーを繋げることによって、どんな新しいサービスでも簡単に短期間に提供可能ということです。

松田：なるほど、いろいろ便利なサービスがこれから提供されそうですが、一方で自動車やドライバーの情報をサービスプロバイダに提供すると、ドライバーの側にプライバシーに対する抵抗が出てくると思うのですが、その点についてどのように考えていますか。

野辺：もちろんプライバシーを守ることは最重要で、目的に応じたことにしか利用しないことを条件にして、ドライバーの同意を得てデータを提供していただいています。でも、意外にドライバーは、自動車会社と通信している以上、自分の車の位置を知っていて当然という意識もあるんです。日産はCARWINGSでコールセンターサービスを提供しているのですが、5～6年前から「この辺でおいしいラーメン屋さんを教えてください」といった電話がかかるようになってきました。実は、プライバシーを配慮して、CARWINGSのオペレータには自動車の位置情報を自動的には知らされていないんです。でも、お客様には自動車会社が位置情報を知っていて当然という価値観があるんですね。位置情報を提供してもいいという人やシチュエーションは意外に多い様です。

エネルギー問題の解決にも期待

松田：原子力発電所の事故によって電力供給事情が厳しくなっています。その点に対しても電気自動車の役割が期待されていますね。

野辺：CARWINGSのサーバーとグリッドのサーバーを繋げば、これからどの辺りの場所でどのくらいのエネルギーを必要とするのかが分かるようになります。そうすると、電力需要のピーク時には充電を避けたり、あるいは太陽光パネル発電から蓄電した電力を後で利用するというように、電気エネルギーの需要と供給をコントロールすることが出来るようになります。また、車のバッテリーを家庭の配電盤に繋げば、車から家庭に電力を供給出来るようになります。今のLEAFは充電を受けるのみですが、LEAFから放電を受ける機能を持つ外部周辺機器の商品化を急いでいます。

松田：今のお話は大事なことですね。蓄電池機能を車に持たせておけば、余った電力を蓄えることが出来、電力が足りなくなったら車から取り出す、といったことが実現出来ます。そうなると、車の役割が大きく変わるように思います。車は基本的には移動の役を担っていますが、エネルギーのシステムと繋がるようになると、停まっていても役に立つわけですね。

野辺：そうです。まだまだバッテリーが高価という問題がありますが、電気自動車が増えればバッテリーの需要が増え、そして需

要が増えれば技術も進化するので、バッテリーの価格や寿命に関する課題も解決に向かうと思います。

松田：LEAFのバッテリーを家庭で使うとどのくらいの時間使えるのですか。

野辺：おおむね2日分です。

松田：大したものですね。

プロプライアタリ技術から汎用技術へ

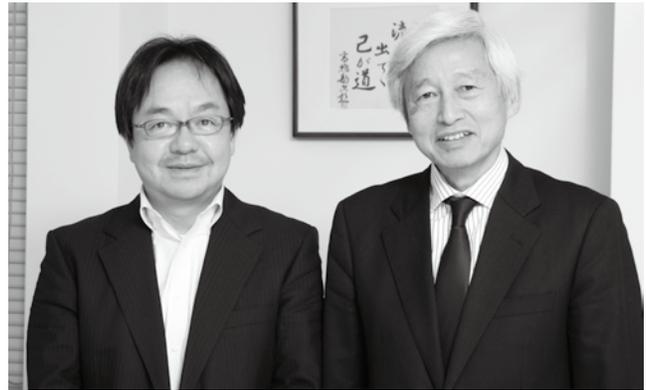
松田：アドバンストサービスを提供していくためのテクノロジーについて伺いたいと思います。

野辺：アドバンストサービスを実現するためには、まさにスマートフォンと同様の技術が車の中に入らないといけないと考えています。あくまでもスマートフォンをそのまま車載化するのではなく、ですが。チップセットもOSもマルチメディアやインターネットと親和性が高くないといけない。従来は、カーナビなど車載用のインフォテイメント的な機器は組込型のプロプライアタリな技術で作り込んでいたのですが、外の世界のインターネットから情報を取るということになると、一般的なスマートフォンで使っているのと同様な技術を取り込むことが基本となります。プロプライアタリな技術で組込みシステムを作り込むと、システムの改変やアップデートは困難です。また、国によって異なる要求を実現するためにはシステムをほとんど作り直さなければなりません。それでは、インターネットやマルチメディアなど、常にアップデートされる情報やサービスをタイムリーに利用することが出来ません。そういう問題に自動車会社も気がついて、プロプライアタリなOSではなく汎用的なOSとしてLinuxを使う自動車会社が増えています。

松田：確かに柔軟にサービスを広げていくためには汎用的なOS上にアプリケーションを作るという構造化が進むと思います。エンタプライズシステムの世界ではプロプライアタリな技術から汎用的な技術へのシフトが起きました。同じことが車載システムの世界でも起こっているのですね。

野辺：インフォテイメントのエリアでは、汎用的なOSで汎用的なソリューションが大事です。汎用化してどの自動車会社でも使うようになれば、数億台の携帯電話のコストに追従出来る可能性があります。そういう視点で考えると、BMWなどが提唱しているGENIVI（ジェニビ）という団体が目指している方向は正しいでしょう。

松田：GENIVIの取り組みというのは？

野辺：GENIVIは、インフォテイメントエリアにおけるリファレンスを作り、そのリファレンスをもとに自動車会社が車載システムを作るという取り組みを進めています。そのリファレンスはLinuxベースです。一方、自動車自体を制御するコントロールエリアは外から操作する必要がありません。コントロールエリアでも、JasParやAUTOSARなど標準化の動きがありますが、それほどジェネラルな標準化ではなく、クローズドした一部の標準化という形で実現すると思います。

松田：今、ITの世界ではシステムに対するアタックが問題になっています。その点についてはどのように考えられていますか。

野辺：車載システムに対するセキュリティ対策は非常に重要だと考えています。ドライバーが見たり操作するインフォテイメントエリアと、自動車の動きを制御するコントロールエリアの間にファイアウォールを置き、インフォテイメントエリアがアタックされても自動車が暴走するといったことがないようにする。自動車を絶対に危険な状態にしない。そういうことをしていかないといけないと思っています。

松田：基本は、インフォテイメントエリアとコントロールエリアの通信をいかに分離するかということですね。

野辺：それと、自動車の車載システムを完全にオープンな方法ではインターネットに直接繋げないことです。

松田：車の側から直接インターネットに繋がるのではなく、1回、別のデータセンターを介するということですか。

野辺：例えばその通りです。自動車会社のデータセンターで繋ぎ先を把握し、ドライバーはそのサービスやコンテンツを利用する形態です。例えば、ドライバーがGoogleマップを利用してPOI（Point of Interest）などの地図情報を検索する場合、日産のグローバルデータセンターは車載機がGoogle MAP Searchに接続する安全な方法を提供しています。

松田：なるほど、車が外部と繋がる部分のセキュリティ対策は厳重にされているのですね。今日は、自動車が無線データ通信によって外部と繋がり、そこにアドバンストサービスを提供する新しいビジネスオポチュニティが生まれてくるというお話を伺いました。そして、ソフトウェアのアルゴリズムやセキュリティ対策など、自動車の未来に対してITが果たす役割は大きいと実感しました。ありがとうございました。

文：小林 秀雄　　写真：越 昭三朗

情報システムの障害状況 2011年前半データ

SEC所長 **松田 晃一**　SEC研究員 **金沢 成恭**

SEC journal No.26で報告した2010年分のデータに引き続き、2011年1月から6月までの半年分の情報システムの障害状況の調査結果を報告する。この間に報道された情報システムの障害は合計9件、月平均1.5件／月であり、これは2010年と同水準である。この期間中には、国民生活に大きな影響を与えた障害が多く発生した。また、原因を見ると、運用や保守における人為的なミスがきっかけとなった障害が5件と目立った。

1. はじめに

私たちの生活に大きな影響を与える情報システムの事故は相変わらず後を絶たない。この状況を少しでも改善するためには、実際に起こった事故の経験を次に生かし、同種障害の再発を防止することが必要である。このねらいでSEC journalでは前号から情報システムの障害に関する情報の連載を開始した［松田 2011］。本稿では、前号で報告した2010年1年間に引き続き、2011年1月から6月までの6カ月間の情報システムの障害状況の調査結果を報告する。

2. 2011年前半の状況

2011年1月から6月までの半年間で報道された情報システムの障害は合計9件、その全体は表1に示す通りとなった。なお、みずほ銀行システムの一連の障害（表1のNo.1105 ）は1件として集計した。障害発生件数を月平均にすると1.5件／月となる。これは2010年の平均値1.42件／月とほぼ同様である。月別の件数を2010年と併せて図1に示す。

2011年前半の障害9件の中には、交通機関に大きな影響を与えた2件、多数の携帯電話に通信障害を発生させた1件、金融機関の長期間に及ぶ混乱1件など、多くの国民の生活に直接多大な影響を与えた情報システムの障害が多く発生したことが特徴である。社会活動や経済活動を支える、いわゆる重要インフラシステムについての障害対策やサービスの早期復旧に関する対策の一層の強化が望まれるところである。

また、全9件のうち原因が報道されている7件について原因別に見ると、ソフトウェア・バグは1件で、残りは運用や保守における人為的なミスが5件、ハード障害1件であり、人為的なミスによる障害が目立つ。もちろん、人為的ミスは障害の直接の引き金となった原因であり、それを引き起こした更に根本的な原因を探る必要がある。例えば、人為的ミスを引き起こさないようなシステム的な対応が開発段階で盛り込まれていれば、ミスを回避出来たと考えられるケースもあるので、再発防止策にはそのような運用や保守段階での作業ミスを回避する対策の検討も重要である。更に、情報システムを長期間運用するうちに環境条件は大きく変化する。開発段階では妥当であった設計条件であっても、時間の経過とともに環境に十分に対応しきれなくなることも起こる。そのような観点での見直しを定期的に行い、システムが環境に適合するよう、必要なシステムの改修や保守など時宜を失わず実施することも重要である。このような観点からシステムを定期的に点検し、必要な処置を講ずることはマネジメントの役割である。

IPA/SECでは、2008年度から「重要インフラ情報システム信頼性研究会」を組織し活動を行い、信頼性を確保するための取組みについての知見を「重要インフラ情報システムの信頼性向上の取組みガイドブック」として

取りまとめた。社会的に重要なサービスを提供している事業者、特にその中の経営層、情報システム部門の幹部や品質責任者が、情報システムの信頼性管理の取り組みを点検するための視点を提供しているので、参考とされたい [IPA2011]。

3. むすび

ここで取りまとめた障害情報は、報道などをもとにSEC において情報を収集し整理したものである。このため、障害の網羅性を保証するものではないが、少なくとも全体の傾向を知る一助になるものと考えている。なお、組込みソフトウェアや海外にセンターを持つと思われる情報システム（例えばクラウドサービス）の事故については、情報が一部に偏る恐れが高いので対象外としている。また、障害の原因については、それを引き起こすきっかけとなった直接的な原因のみにとどめ、その背景にあるより根本的な原因の分析はしていない。この連載の目的は、同種障害の再発防止による情報システムの信頼性向上の一点にあり、発生した障害に対する責任を追及したり、ましてや特定の組織を非難する意図は全くないことは言うまでもない。

このような情報の開示が一般的に行われるようになり、より深い原因分析と再発防止策の共有によって、情報システムの障害が減り、より一層安心・安全な情報社会がもたらされることを期待したい。

参考文献

[松田 2011] 松田晃一 , 金沢成恭：情報システムの障害状況　2010 年データ , SEC journal No.26, Vol. 7, No3, pp.102-104, Oct.2011
[IPA 2011] IPA/SEC：重要インフラ情報システムの信頼性向上の取組みガイドブック～情報システムの信頼性管理に必要な組織内の役割分担と活動の枠組み～ , http://sec.ipa.go.jp/reports/20110330/20110330.pdf, Mar.2011

図1　情報システム障害の月別発生件数（報道に基づきSECが整理）

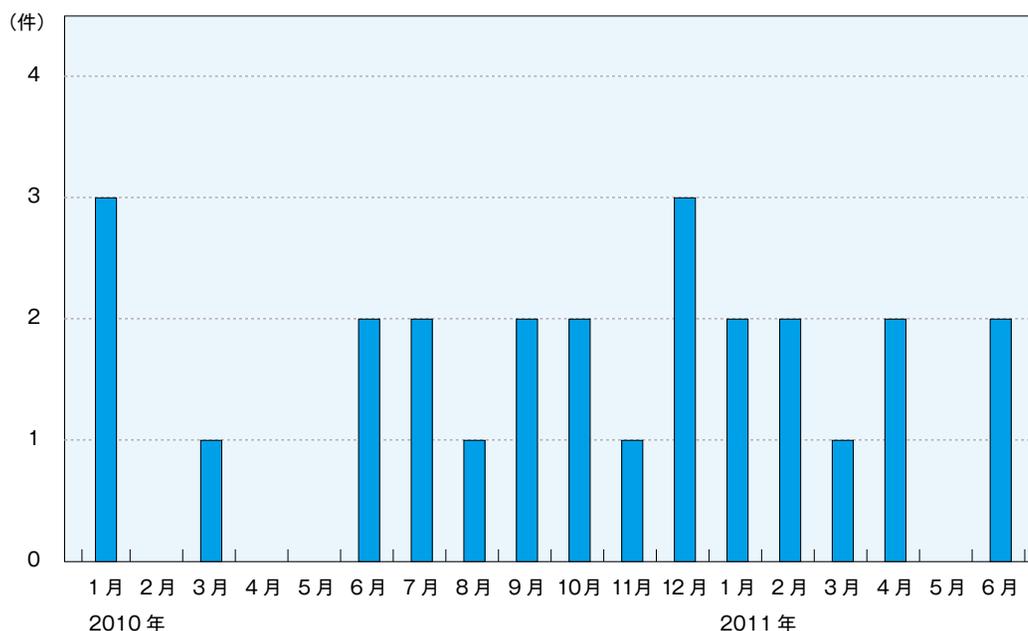



表1　2011年前半の情報システム障害データ（報道に基づきSECが整理）

No.	システム名	発生日時（上段）回復日時（下段）				影響	現象と原因	直接原因	主な情報源
		年	月	日	時				
1101	三菱UFJ信託銀行システム	2011	1	4	8時00分	オンラインシステムに障害発生。全国の本支店の店頭及びATMでの入出金、為替、照会などの取引2,884件が不能に。インターネットバンキング利用の988名が取引出来ず。ゆうちょ銀行やコンビニ店などでのATMにおける2,805件の取引が出来ず。	1日から3日にかけて実施したシステムの更新作業で、更新すべきファイルを取違えたため。	保守時の人為ミス	・三菱UFJ銀行報道発表（2011.1.4.5）
		2011	1	4	11時半頃				
1102	JR東日本新幹線システム	2011	1	17	8時23分	新幹線の運行管理システム（COSMOS）においてダイヤの変更入力を行った際、予想ダイヤが表示されなくなったため、確認のため全列車を停止させた。列車8本が立ち往生。運休・遅延本数は139本、8万1,200人に影響。	新幹線の運行管理システム（COSMOS）において、ダイヤ変更入力時に、修正データ数がシステムの限度値600件を超えると、予想ダイヤを表示出来ない実装となっていた。このことに関する情報共有が出来ていなかった。	システムの限界値を超えて入力したミス	・JR東日本報道発表（2011.1.18）
		2011	1	17	9時38分				
1103	証券保管振替機構ゲートウェイシステム	2011	2	4	9時00分	“ほふり”のゲートウェイシステムに障害が発生し、日銀金融ネットワークに接続が不能となった。このため当日を決済日とする一般債、短期社債、投資信託の決済処理が出来なくなった。端末からのデータ入力の代替措置により決済は完了させた。	ゲートウェイシステムの認証にかかわる設定の誤り。	設定誤り	・証券保管振替機構報道発表（2011.2.4）
		2011	2	7	9時00分				
1104	日本年金機構ホームページ	2011	2	28	9時00分	2月28日より新たに「ねんきんネット」サービス（年金加入記録をインターネット経由で個人が照会出来るサービス）が開始されたが、年金機構のホームページからログイン出来ない事態が発生。代替の方法を案内（ログイン用のURLを直接入力するなど）。	新サービスのプログラム設定ミス？	―	・日本年金機構報道発表（2011.2.28）
		2011	2	28	22時頃				
1105	みずほ銀行システム	2011	3	15		オンライン取引の開始が大幅に遅延。一部の取引はその後も利用出来ず。前日の夜間バッチ処理の異常終了により、決済処理が約38万件未処理となる。	東日本大震災の義援金振込みが、特定の支店口座に集中し、あらかじめ設定してあった夜間バッチ処理の一口座当たりの処理件数の上限値を超えたため、夜間バッチ処理のエラーが多発し、処理の大幅な遅延を招いた。	運用ミス	・みずほ銀行　システム障害特別調査委員会調査報告書（2011.5.20） ・日経コンピュータ 2011.4.28 号 動かないコンピュータ ・日経コンピュータ 2011.6.9 号 みずほ銀障害の全貌
		2011	3	15	10時25分				
		2011	3	16		オンライン取引開始処理が大幅に遅延。一部時間帯にATMサービスやインターネットバンキングなどが利用停止となる。振込み約44万件、総額約5,700億円の手続きが未処理。	前日の夜間バッチが完了せず、更にATM障害が発生し障害対応も必要となり、取引開始が大幅に遅延。異常終了した夜間バッチ処理の回復が出来ず、新たな取引の処理も未処理となる。		
		2011	3	16	11時12分				
		2011	3	17		16日と同様、オンライン取引開始処理が大幅に遅延。一部時間帯にATMサービスやインターネットバンキングなどが利用停止となる。振込み約50万件の手続きが未処理となる。	異常終了した夜間バッチ処理が夜間の時間帯中に回復出来ず、更にATM障害も重なり、オンライン取引開始が大幅に遅れた。新たな取引のバッチ処理も持ち越され未処理となる。		
		2011	3	17	10時46分				
		2011	3	18		給与振込み約62万件、総額1,256億円が処理出来ず。決済の未処理は17日までの約50万件と合わせて約112万件となった。店頭で仮払い。ATMは預金の出し入れに限って稼動。インターネットバンキングは停止。	異常終了した夜間バッチ処理の回復が出来ず、新たな取引の処理も未処理となる。		
		2011	3	18	10時頃				
		2011	3	19		3連休（19～21日）中はATM、インターネットバンキングは完全停止。22日朝、振込み以外の業務は正常に回復。振込みを含め全業務は22日午後回復。しかし、未処理振込みが更に残っていることが判明、24日すべての処理が完了した。	オンライン業務を停止し、夜間バッチ処理の未処理分の回復を図り22日時点で大半が解消した。しかし、作業時間の不足により22日時点で約16万件が未送信、23日も約1,000件が未送信となり、すべての解消は24日となった。		
		2011	3	24					
1106	ゆうちょ銀行システム	2011	4	9	8時45分頃	東日本地域（北海道、東北、関東、信越など）の約1,000台のATM（全国ATM約26,000台の内約4%に相当）で、取引出来ず。	7日夜に発生した地震による停電からの復旧作業における人為的ミス。	人為ミス	・日本経済新聞（2011.4.10 朝刊） ・ゆうちょ銀行報道発表（2011.4.9.10）
		2011	4	10	朝				
1107	新生銀行システム	2011	4	25	10時14分頃	個人向け、提携先など約65,000台のATM及びインターネットバンキング、モバイルバンキングのサービスが停止。	顧客情報や取引データを保存したデータベースの障害（ハードかソフトかは不明）。	―	・新生銀行　報道発表（2011.4.25）
		2011	4	25	11時36分				
1108	NTTドコモシステム	2011	6	6	8時27分	関東・甲信越の約172万台の携帯電話の通信障害。	携帯電話の位置情報を管理するシステムのパッケージ（ハードウェア）故障がトリガーとなり、システム切り替えが発生し、位置登録の負荷が急増。ソフトウェアの過負荷耐性が不足していたため、システムの処理能力が低下しふくそう状態となったため、通信を規制した。システムが安定したため通常状態へ移行したところ、切替ソフトウェアの不具合により、再度システム切替が発生し、同様の事象が再発した。	ハード障害によりソフト不具合が顕在化	・NTTドコモ　報道発表（2011.6.14） ・日経コンピュータ 2011.7.21 号
		2011	6	6	21時36分				
1109	国土交通省航空交通管理センターシステム	2011	6	16	4時頃	航空各社から提出された飛行計画を一括集約して全国各地の航空管制に配信する飛行情報管理システムの障害。43便に30分以上の遅れ。最大1時間22分の遅れ。	電源装置が故障し、バックアップ機も使用出来ず。電源装置の交換により復旧。その間、ファックスによる手作業で対応。	電源障害	・日本経済新聞（2011.6.16 夕刊）
		2011	6	16	7時前				

SEC journal 論文賞
受賞論文発表

SECは、我が国のソフトウェア産業発展のための様々な取り組みを実施しておりますが、その取り組みの1つとして、ソフトウェア・エンジニアリングに関する論文に賞を設け、表彰を行っております。

今回のSEC journal論文賞は、2009年12月から2011年6月までに投稿された合計9編のうち、査読者により採録された4編の論文を候補とし、選考委員会による厳正な審査の結果、3編を表彰候補論文として選出いたしました。

各賞の決定と発表は、IPAフォーラム2011（2011年10月27日）において、SEC journal論文賞表彰委員会によって行われ、今回は優秀賞3編が表彰されました。片山委員長による審査報告は155頁に掲載されています。なお、3編の優秀賞受賞論文は22号、24号、26号に掲載されています。

SEC journal 論文賞表彰委員会

委員長	片山　卓也	北陸先端科学技術大学院大学　学長
委員（50音順）	有賀　貞一	AITコンサルティング株式会社　代表取締役 （株式会社ミスミグループ本社　前　代表取締役　副社長）
	井上　克郎	大阪大学大学院　情報科学研究科　教授
	大原　茂之	東海大学　専門職大学院　組込み技術研究科　教授
	鶴保　征城	学校法人・専門学校HAL東京　校長
	松本　健一	奈良先端科学技術大学院大学　情報科学研究科　教授
	松田　晃一	独立行政法人情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター　所長

SEC journal 論文賞選考委員会出席者

委員長	松本　健一	奈良先端科学技術大学院大学　情報科学研究科　教授
委員（50音順）	飯泉　紀子	株式会社日立ハイテクノロジーズ　研究開発本部　第四部　主任技師
	片岡　欣夫	株式会社東芝　研究開発センター　システム技術ラボラトリー　研究主幹
	二上　貴夫	株式会社東陽テクニカ　ソフトウェア・システム研究部　部長
	古山　恒夫	東海大学　理学部　情報数理学科　教授
	山城　明宏	東芝ソリューション株式会社　ソリューション技術統括部　主幹
	山本　修一郎	名古屋大学　情報連携統括本部　情報戦略室　教授
	山本　雅基	名古屋大学大学院　情報科学研究科　ディレクタ・特任准教授
	山本　里枝子	株式会社富士通研究所　ソフトウェアシステム研究所　シニアディレクター
	新谷　勝利	独立行政法人情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター
	松田　晃一	独立行政法人情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター　所長
	三原　幸博	独立行政法人情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター
	山下　博之	独立行政法人情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター

優秀賞

Eメールアーカイブのクラスタリングによる開発コンテキストの可視化

大蔵 君治，川口 真司，飯田 元

特定デザインパターンに基づく大規模基幹システムのオープン化技法

北川 陽一

CoBRA法を使った見積りモデル構築のポイント

酒井 大

上段左より、新谷勝利・立石譲二・山下博之・三原幸博
鶴保征城・ポール E. ブラック・有賀貞一・井上克郎・松本健一・大原茂之
松田晃一・大蔵君治・北川陽一・酒井大・片山卓也・藤江一正

（敬称略）

SEC journal 論文賞　表彰委員会審査報告

SEC journal 論文賞
表彰委員会委員長
北陸先端科学技術大学院大学　学長
片山　卓也

SEC journalでは、ソフトウェア開発現場のソフトウェア・エンジニアリングをテーマとした実証論文を掲載してきました。単なる理論的な研究ではなく、現実に使える手法や実践経験など開発現場における貴重な知見を報告し、それにより新しい手法や方法論の導入促進に大きな役割を果たしてきました。具体的には、①開発現場への適用を目的とした手法・技法の詳細化・具体化などの実用化研究の成果に関する論文、②開発現場での手法・技法・ツールなど様々な実践経験とそれに基づく分析・考察、それから得られる知見に関する論文、③開発経験とそれによる現場実態の調査・分析に基づく解決すべき課題の整理と解決に向けたアプローチの提案に関する論文の分野です。このような分野の論文を開発現場や大学・研究機関などから広く投稿を求め掲載してきました。表彰委員会では、毎年それらの中から優れた論文を決定し、論文賞を授与してきました。

今回は投稿された9編の論文を、奈良先端科学技術大学院大学教授松本健一委員長を中心とする論文選考委員会において厳正に査読を行い3編の論文賞候補を選出し、それをもとに表彰委員会が授賞論文を決定しました。表彰委員会では、各々の論文の著者にプレゼンテーションをしていただき、質疑応答を行い、表彰論文に相応しいことを確認すると同時に、賞（最優秀賞、優秀賞、所長賞）を決定しました。その結果、今回は以下に紹介する3編に優秀賞を授与することを決定しました。

(1)「Eメールアーカイブのクラスタリングによる開発コンテキストの可視化」大蔵君治，川口真司，飯田 元

ソフトウェア開発における失敗や納期の遅れの分析は、必ずしも十分には行われていないのが現状である。プロジェクトの実行記録が十分に取られていないことや、終了プロジェクトに対してコストをかけて分析を行う余裕が開発現場に無いことが大きな理由である。本論文は、開発メンバー間で交わされるメールを系統的に分析することにより、問題点の把握を行う分析法を提案したものである。特徴語によるメールのベクトル化と類似度解析・クラスタ化に基づく方法を提案し、その有効性を実プロジェクトデータに適用して示している。提案されたメールの分析手法と同時に、開発中に特別に記録を取る必要がない現実的な方法であることが高く評価された。

(2)「特定デザインパターンに基づく大規模基幹システムのオープン化技法」北川陽一

基幹情報システムがレガシー化する中、その若返りや再構築は社会的に重要な課題である。現状の機能を正しく維持しながら、将来の変更や高度化に備えた進化性の高いシステムを再構築する必要がある。本論文は、証券基幹システムの再構築に際して、オンライン処理システムをオブジェクト指向方法論によりオープンシステムを設計する際、特別に開発されたRSU（Root, Stage, Unit）と名付けた特定デザインパターンを適用することにより、処理内容の記述を明快に行い、期間や工数を予想より大幅に短縮することが出来たことの報告である。新しい手法を実プロジェクトに適用し、その効果が定量的に評価・確認されており、SEC journalに相応しい優れた論文であることが評価された。

(3)「CoBRA法を使った見積りモデル構築のポイント」酒井 大

開発工数の確度の高い見積りは、開発プロジェクトを成功させる大きな要素である。工数見積りのためのモデルや方法論はいろいろと提案されているが、現実には的確な工数の見積りは簡単ではない。一方、優れたPMの経験と少ないプロジェクトデータから説明性の高い工数の推定が可能な方法としてCoBRA法が知られており期待が高いが、その適用にはパラメータの推定などに関してノウハウが必要である。本論文は、著者の属する組織における主に変更開発にCoBRA法を適用した経験を述べたものであり、貴重な報告である。CoBRA法を実践する上での様々な工夫が示されており、現場での適用に大いに参考になるものであることが高く評価された。

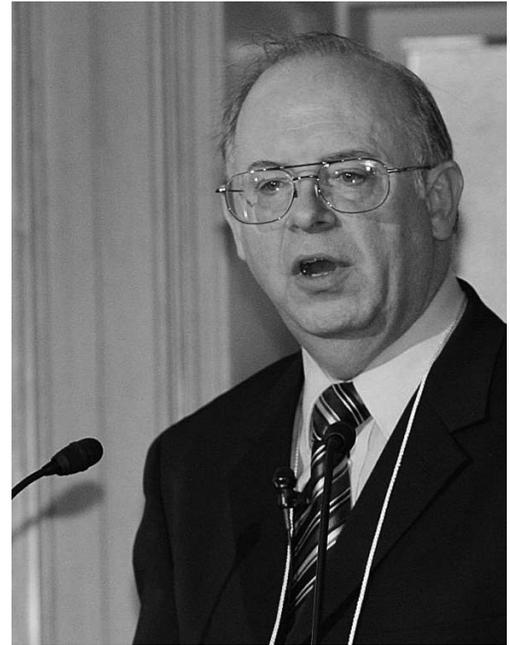
最後に、今回論文賞に選ばれた3編はいずれも優れた内容のものであり、ソフトウェア開発現場やソフトウェア工学研究コミュニティにとって大変貴重なものである。

SEC journalの発行の趣旨を鑑みると、今後とも広く論文を募集して、開発現場での経験や実践的な研究成果を掘り起こすことが求められている。我が国のソフトウェア開発力の向上には、ソフトウェア工学の学術的な研究と同時に、現場での評価や経験が極めて重要である。SEC journalはこのような情報を産業界の技術者に直接届けることが出来る貴重なメディアであり、今まで以上に充実した内容を期待している。

ソフトウェアの品質保証とテスト

～メトリクスと測定、明確な記述、そして管理可能なプロセスの三本の柱がより信頼できるソフトウェア開発に寄与する～

ポール・イー・ブラック博士
NIST（米国商務省国立標準技術研究所）
ITL（情報技術研究所）、SSD（ソフトウェア＆システムズ部）

SEC調査役
新谷 勝利

はじめに

SECでは、その開設から今日までSEI※1及びIESE※2との共同研究を推進している。加えて、2010年からは、欧米の政府関連ソフトウェア機関、すなわち、NIST※3及びCEA-LIST※4に拡大している。IPAフォーラム2011にあたり、SECの今年度の強化方針の一つの「ソフトウェア品質の説明責任の制度化」に関連する技術動向の講演をNISTにお願いした。本稿は、IPAフォーラム2011（2011年10月27日）における講演を編集したものである。

概要

1　NISTとは？
2　Combinatorial testing（組み合わせテスト）
3　SAMATE※5について
4　バグを数えるのは難しい
5　Software assurance（ソフトウェアの品質保証）について
6　ソフトウェア品質保証を達成する3本の柱
7　おわりに

1 NISTとは？

今回は、我々がソフトウェア保証という分野で何をやっているかというお話をさせていただきます。その前に、図1にある我々のNISTという組織は、準拠しなければいけない、また、準拠すべきであるという様々な基準を作っています。古くは度量衡です。これは規制当局ではありません。法律を作っているのではないのです。NISTには3,000名ほどの専門スタッフがおり、我々自身が研究活動を行っています。基準を作る、標準化を推進する等いくつもの分野があります。例えば、歯科用のセラミックの研究、文書検索、DNAを活用する法医学、バイオメトリクスといった分野があります。ソフトウェアというのは、その中の一分野です。私は、ソフトウェアを担当するITL※6の図2にあるSSD※7に所属しています。SSDには、いろいろなグループがあり、プロジェ

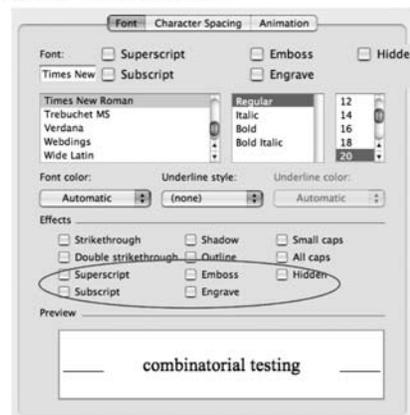
図1　NISTとは

図2　SSD（ソフトウェア&システムズ部）

図3　ワープロ飾り文字の選択に関する考察

クトごとに連携をとっています。今回はソフトウェア品質保証にかかわるいくつかのプロジェクトについてお話をします。ソフトウェアの品質を保証するためには、「スマートなプログラマが必要なだけではないか」とおっしゃる方もいるかもしれません。でも、我々が求めているのは、ヒーローがやってきて、コードを書き、問題を片付けてくれる（これはカウボーイ・コーダと称されています）というものではありません。チームワークが大事なのです。我々が目指している品質の高いソフトウェアを作り出すためには、チームワークが必要なのです。これから、チームで実施している種々のプロジェクトについて、お話をしていきます。

|2| Combinatorial testing（組み合わせテスト）

テストでは、品質の確認すべてを出来るわけではありません。しかしながら、テストは今まで品質保証のために重要な位置付けを持っていました。どの程度のテストが出来るのかに関して、これまでいくつかの研究により、ほとんどの場合において、バグは2つから3つのパラメータによって引き起こされる相互作用によるということが分かってきています。医療機器、NASAのソフトウェア、コンピュータのサーバ、ブラウザ等、こういった分野でのテストの実際を検討してきました。結果、6つの相互作用が起こっているということも分かっています。6つまでテストでカバーすると全部のバグを発見出来るということになります。ではテストによってこのような相互作用をどのようにカバーしていけば良いのか、幾何学的に捉えていきたいと思います。

例えば図3で示すワープロ用ソフトウェアが目の前にあるとします。いろいろな文字飾りをonにすることが出来ます。ここでは図3の丸囲みの5つだけを使うとしましょう。Emboss（浮き出し）はonにしたりoffにしたりすることが出来る一次元の問題となります。そしてEngrave（浮き彫り）を足しましょう。そうすると二次元になって4つのテスト項目ということになります。そして今度はHidden（隠し文字）で、三次元のものになります。そうすると8つのものになりますね。そして Superscript（上付き文字）、Subscript（下付き文字）を足

脚注

※1　SEI：Software Engineering Institute, カーネギーメロン大学ソフトウェア工学研究所
※2　IESE：Institute for Experimental Software Engineering, ドイツ・フラウンホーファ財団実験的ソフトウェア工学研究所
※3　NIST：National Institute of Standards and Technology, 米国商務省国立標準技術研究所
※4　CEA-LIST：フランス原子力・代替エネルギー庁システム統合技術応用研究所, Commissariat a l'Energie Atomique et aux Energies Alternatives, Laboratoire d' Integration des Systemes et des Technologies
※5　SAMATE：Software Assurance Metrics And Tool Evaluation
※6　ITL：Information Technology Laboratory, 情報技術研究所
※7　SSD：Software and Systems Division, ソフトウェア＆システムズ部

Why? A Geometric Intuition

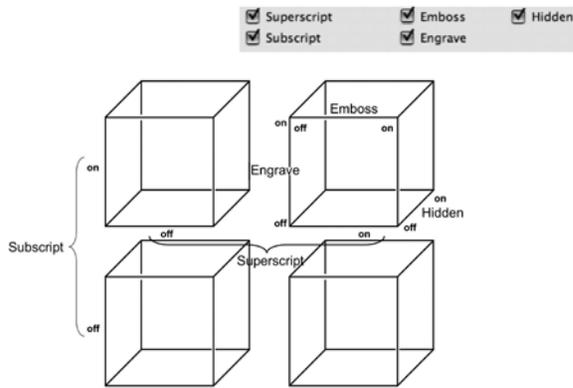



図4 選択に関する幾何学的直感

Naïve Test Approach is Sparse

- Test all off, all on, each one on
  - 7 tests total

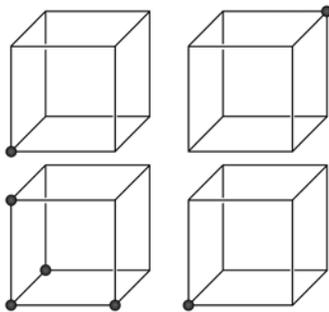



図5 テストアプローチを空間的に観る

How Many Tests Does it Take?

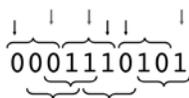
- There are $\binom{10}{3}$ = 120 3-way interactions.
- Naively 120 x 2^3 = 960 tests.
- But each test exercises many triples:

We can pack many triples in one test, so what's the smallest number of tests we need?

図6 何回テストすれば良いのか?

していくと五次元になっていくわけです。これは図4のように幾何学的に表現出来ます。

ここで、これをどうやってテストするかということを考えてみましょう。図5で示すようにすべてのオプションをoffにして、それからすべてのオプションをonにするケース、そして1つだけonにしていくというケースを考えます。

その組み合わせの中で、それがどうonになるか。全部offにしますと左下に示すようになります。すべてのオプションをonにすると右上になります。そして、各コンビネーションで1つずつonにいたしますと、トータルで図5のように7つのテストになるわけです。しかしながら、すべての組み合わせをカバーしていないということが分かります。組み合わせテストというのは、全空間を使っていくようにマッピングを考え、有効な形ですべてのテストを行っていくことが出来るようにする方法です。

より大きな例を考えてみましょう。10の飾り文字があり、10を全部使っていくとしましょう。飾り文字間の相互作用というのがあります。3-wayの相互作用は$_{10}C_3$で120の組み合わせがあります。上付き文字、下付き文字、隠し文字といったものを全部考えていった場合、図6のように、この全部の組み合わせは960のテストということになります。

しかし、1つのテストではいろいろなことが出来るのです。図6のトリプル（三つ組）を捉えて1つの三つ組をテストして、それから非常に多くの三つ組をいっぺんにテストすることが出来るわけです。そして異なる様々な三つ組を1つのテストの中に押し込むことが出来るわけなのです。そうすると、すべてのコンビネーションを確かめるために、10の変数が散らばっている形にするにはどうしたら良いのか？ 何回のテストが必要か？ ということになるのですが、それを考えると図7に示すように13回で済むことになります。

このチャートはそれぞれの相互作用を表しています。上付き文字、下付き文字、そして隠し文字といったものが1つひとつになっています。そうすると、この三つ組

All Triples Take Only 13 Tests!

0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	0	0	0	1
1	0	1	1	0	1	0	1	0	0
1	0	0	0	1	1	1	0	0	0
0	1	1	0	0	1	0	0	1	0
0	0	1	0	1	0	1	1	1	0
1	1	0	1	0	0	1	0	1	0
0	0	0	1	1	1	0	0	1	1
0	0	1	1	0	0	1	0	0	1
0	1	0	1	1	0	0	1	0	0
1	0	0	0	0	0	0	1	1	1
0	1	0	0	0	1	1	1	0	1

図7　三つ組みの組み合わせテストでは13回で十分

What is the SAMATE Project?

- Software Assurance Metrics And Tool Evaluation (SAMATE)
- It is sponsored in part by U.S. Department of Homeland Security
- Current areas of concentration
  - Web application scanners
  - Source code static analyzers
  - *Static Analysis Tool Exposition (SATE)*
  - *Software Reference Dataset (SRD)*
  - Research into software assurance and tools
  - *Software labels*
  - Malware research protocol

SAMATE

http://samate.nist.gov/

図8　SAMATEプロジェクトとは?

については８つのテストで全部カバー出来ます。次の三つ組を考えてもこうなります。そしてこの三つ組のカラムはこれだけで全部の組み合わせがカバーされる形になるわけです。非常にリーズナブルなコストで、つまり、かなりの小規模なテストですべての組み合わせをテストすることが出来るということになります。

これらについて興味のある方は、このNISTのサイト[※8]から無償のソフトウェアをダウンロードして、皆さんのマシンでテストしていただくことが出来ます。

|3| SAMATEについて

次に、SAMATEのプロジェクトについて説明します。7つの分野が図８に示されていますが、その内の２つ、ソフトウェア・ラベルとSATE[※9]について説明します。

●ソフトウェア・ラベルについて

情報を交換するために、すなわち、ユーザに対して情報を伝えるために、対象のソフトウェアはどういうものかを認識出来るものが必要なわけです。ソフトウェアの特性だけではなく、図９に示すような考え方で、そのラベル付けを行っていくという働きかけを始めました。このプロジェクトでは、どういった情報を提供出来るのか、

Software Labels

- Software should list facts like a nutrition label or material safety data sheets (MSDS)
- Like food, it does not say everything about the software, but gives key content, e.g.,
  - Is default installation secure?
  - Accessed: network, disk, …
  - Certificates
- One step toward market for better software
- Cautions: labels may
  - Give false confidence,
  - Shut out better software, or
  - Divert effort from real improvements.

https://www.aspectsecurity.com/

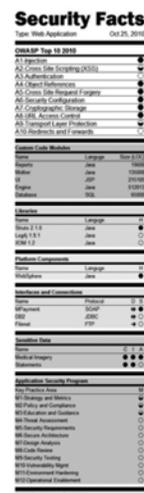



図9　ソフトウェア・ラベルとは

脚注

※8　http://csrc.nist.gov/
※9　SATE : Static Analysis Tool Exposition

どういう情報を提供すれば本当に役に立つのか。そして、消費者、ユーザは何を必要としているのか等を検討しています。

● SATE について

この図10で示すSATEのプロジェクトにおいては、NISTは、セキュリティ関連のプログラムを用意し、参加組織はツールを実行します。そして結果を我々にまた送り返してもらいます。我々は送られてくる報告書から、どのツールがどういったものを見つけたか、ということをまとめます。ここでは、最良のツールを決めることが目的ではなく、どのツールがどのように改善出来るか等の情報を共有するのが意図です。静的解析ツールを使っていくことを推奨していくものです。皆さんがもし静的解析ツールを作っていらっしゃるということであれば、将来的にこれに参加していただければ良いと思います。

現在は、SATE Ⅳの段階ですが、既に一連のプログラムを用意しました。そして、参加者に対してはこのプログラムを7月末に提供しました。そしてこれから解析を行い、2012年の3月30日にはワークショップを開きます。

Static Analysis Tool Exposition
SATE

- Events
  - We choose programs with security implications
  - Participants run tools and return reports
  - We analyze reports
  - Everyone share observations at a workshop
  - Release final reports and all data later
- Goals:
  - Enable empirical research based on large test sets
  - Encourage improvement of tools
  - Speed adoption of tools by objectively demonstrating their use on real software
- NOT to choose the “best” tool

図10　SATEとは

| 4 | バグを数えるのは難しい

率直に申し上げたいのは、バグを数えるのは難しいということです。プログラマはコードを見るときに「どのくらいの数のバグがあるのかな」「リリースして、この数でも大丈夫かな」と考えるわけですが、これはとても難しい問題です。例えば、侵入者が攻撃してしまったら重大故障になってしまうかもしれない、サーバにダメージを与えるかもしれない、ということも考えています。攻撃の規模が分からなければ、どう対処したら良いかが分からない。全く使われないコードがあったとして、それはわざわざ直すべきなのかどうか、使われもしないのに直すべきなのかという問題もあります。

次に、バグに関する具体的なお話をいくつかしたいと思います。

例えばStringを使用するときには、バッファの長さを指定します。Stringの長さが想定したバッファに入るものであれば問題はありません。しかし、Stringが大きくなりバッファに入りきらなくなると問題が発生します。そうならないようにするには、Stringとバッファの長さをチェックし、例えばバッファoverflowというフラグを立て、それに応じた処理とするようにします。

このようなことを考慮したコードを作成することによって、バグが出ないようにすることは出来ます。しかしながら、とても使いにくく、分かりにくいコードになってしまいます。

次に言語標準の場合です。例えば図11のように10字分が割り当てられています。ほとんどのコンパイラはラウンドアップすることによってエクストラ・スペースを確保することが出来ます。ここで、この「10」のところに文字を置くことが出来ます。「0」から「9」まで数えていくわけですが、「10」というのはそこから外れてし

まうわけです。しかし、コンパイラとしてはそのスペースを確保するか、エラーとするかどうか、我々としては懸念する必要はないかもしれないわけです。ですから、こういったところで問題かどうかということもバグの1つの考慮点と言えましょう。

そして、インプット時に、「0」で割るという場合があります。「これはエラーではない」という人もいるわけです。つまり、IEEE規則においては、浮動小数点数という非常に具体的に答えを出しています。つまり“not a number ?（数ではない）”と規定しているのだから問題ではないと。エラーなのか、それともそうではないのかと定義をすることは非常に難しいという問題になるわけです。

また、何回も繰り返されるミスというものがあるときに、コードには*N*個のバグがあるというのか、*N*個の回数のバグが観測されるというのか、という問題もあります。

更に脆弱性のクラスについて考えてみますと、例えばSQL Injectionがあります。これがユーザの方の問題、つまり、SQLエンジンの方で何か予期しないことをやってしまった、そしてCommand Injectionも同じようなものでしょう。それからScriptingも同じです。これは一つのタイプのweakness、つまり不適切なInput Validationというものに起因しています。つまり、別のツール、そしてそれぞれのチェックがImproper Input Validationを行っています。一つは正しいし、もう一つも正しい。つまり、両方が起こしている可能性があるわけです。

すなわち、バグではないと正当化するルールがないとバグと決め付けることは出来ません。こういったケースはたくさんあります。つまり、実際にバグを数えることは難しいということになります。

| 5 | Software assurance（ソフトウェアの品質保証）について

ソフトウェア品質保証の要素は何か。

その要素というのは図12にあるように3つあると思っ


```
typedef struct {
  int int_field;
  char buf[10];
} my_struct;

int main(int argc, char **argv){
  my_struct s;

  s.buf[10] = 'A';

  return 0;
}
```

図11　バッファオーバーフローの例

図12　ソフトウェア品質保証の要素

ています。

まずは、p、開発プロセスの品質です。つまり、きちんとした要件をちゃんと収集しないと、そしてスペックもきちんと修正しないとソフトウェアというのはきちんと出来上がりません。つまり、ソフトウェアの開発プロセスが行われていないと、そして開発がそれを考慮しないと、ソフトウェアを作ることは出来ないということになります。

次いで、s、ソフトウェアの評価された品質です。第三者機関で、ソフトウェアに対してテストをするわけです。とても良いプロセスがあり、高い評価を受けていれば、テストはそれほどに必要ないかもしれません。もし、プロセスの実践が明確ではない、ということであれば、テストが非常に必要になるでしょう。

次に挙げるのは、e、実行時の回復力です。システムとしてバグに対して回復力が高い、エラーに対して回復力が高いということになれば、品質は高いということになるわけです。

ソフトウェアの品質保証には、この三要素が絡まってくる。これらがお互いにサポートしあうことになります。

アセスメント（評価）というものもアシュアランス（保証）の基本要素になります。2種類の分析がアセスメントにおいてなされます。静的分析と動的分析です。動的というのはいろいろな種類のテストで、静的というのはコードレビュー、統計的な分析、モデルチェッカー、アシュアランス・ケース等です。

これら2つのものはそれぞれ補完的に働きます。

| 6 | ソフトウェア品質保証を達成する3本の柱

最後に3つの柱ということですが、このソフトウェア保証を次の段階に進めるためには何が必要かということです。私が提案したいのは、まず、メトリクスと測定するということ。次いで明確に記述するということ。例えば、非機能要件を明確に仕様記述するということ。そして、ソフトウェア構築にあたり、それを管理可能なプロセスにするということです。この3つを適用することによって、より高いレベルのソフトウェア保証を行うことが出来ます。コンピュータ・サイエンスにより、ソフトウェア一般というのは、全部を測定するということは不可能であるということが分かっています。

| 7 | おわりに

私たちを取り巻く社会は、以下の選択肢を持っていると考えます。

- ・ソフトウェアによる不具合は仕方のないものと受け入れる。
- ・ソフトウェアのサイズや権限を限定していく。
- ・ソフトウェアの問題点を解決し、そしてしっかりと機能するものにする。

みんなで協力をしていけば、すべてのソフトウェアをちゃんと機能するものにしていくことが出来るものと信じています。

ご清聴ありがとうございました。

ソフトウェアの品質説明力強化の取り組み

SEC 統合系プロジェクト
研究員
田中　和夫

電気やガスの供給、ネットバンキングなどの社会基盤は、ソフトウェアがサービスの実現に大きな役割を担っており、ソフトウェアのバグ（欠陥）等によって障害が起こると、多数の消費者に大きな影響を及ぼす。本稿では、ソフトウェアの品質に第三者がお墨付きを与え、消費者に分かりやすく示すことにより、消費者が安心して製品やサービスを利用できると共に、国際的にも通用することを目指したソフトウェア品質監査制度（仮称）のフレームワーク案を紹介する。

1　はじめに

昨今、ソフトウェアが組み込まれた機器やソフトウェアで実現するサービスは国民生活においてなくてはならない社会基盤になっている。私たちが日常欠かすことの出来ない電気やガス、水道の供給をはじめ、金融、銀行システム等の重要インフラ等システム[※1]では、ソフトウェアがサービスの実現に大きな役割を担っている。

また、このような社会活動の基盤として機能するシステムの多くは、本来独立に開発・運用されてきた組込みシステム製品（携帯電話や家電等）や情報システム（予約システム、ネットバンキング等）が相互に有機的に連携することによって、全体として新たな機能を果たす形態（以下、統合システム）が当たり前になってきた（図1）。

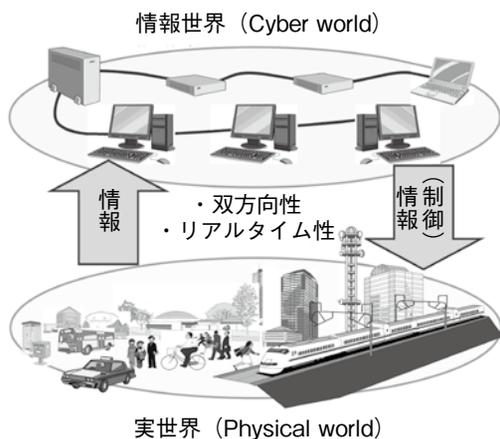



図1　統合システムの例（CPS[※2]）

しかし、消費者への利便性が向上する反面、私たちは潜在的な危険にさらされている。SEC journal No.26[SEC journal 26]では、2010年に発生したシステム障害状況を掲載している。その中には、ソフトウェアのバグ（欠陥）等により航空機の管制システムの障害が発生し、航空機が運休する事例や、銀行のオンラインシステムの障害が発生し、ATM等でキャッシュカードの取り扱いが出来なくなる等、私たちに大きな影響を及ぼす事例が多い。

例えば、2010年7月に発生した、ゆうちょ銀行システムの障害では、全国約2万6千台のATMで他行のカードを使った取引や他行への送金が出来なくなり、インターネットバンキングでも他行への送金が不能になるなど約1万件の取引に影響した。

このような障害は、機器・サービスを提供する事業者の信用低下をまねき、消費者の不信や不安を引き起こす原因となる。

2　ソフトウェア品質説明力強化の必要性

消費者の不安を無くし、事業者の信用を向上させるた

脚注

※1　重要インフラ等システム：経済産業省「情報システムの信頼性向上に関するガイドライン」（平成18年6月15日）の中で定義されている。その信頼性が国民生活または社会経済活動に多大なる影響を及ぼすシステムを指し、10分野に分類されている。

※2　CPS：Cyber Physical Systems, 物理的な設備・機器等とソフトウェアが連携して機能するシステム

めに、製品やシステムが提供するサービスの信頼性・安全性をどのように担保し、説明したら良いだろうか。また、その説明は当事者である事業者からだけで十分だろうか。

例えば、2010年初めに米国で自動車の電子制御システムの安全性に関する疑惑が生じたが、米国では、事業者の説明だけでは不十分とされ、公的機関が検証・妥当性確認を行った※3。

この対応を契機に、我が国でも組込みシステムの信頼性を第三者が客観的に立証する仕組みの整備が急務とされ、経済産業省の産業構造審議会で、ソフトウェア品質に関する第三者による検証・妥当性確認のフレームワークの必要性が示された。

そこでSECでは、ソフトウェアの品質を利用者に客観的に示す枠組み検討のため、ソフトウェア品質監査制度（仮称）の制度化に向けた部会を設置して、2010年11月から2011年6月にかけて計7回開催し、2011年9月30日に「ソフトウェアの品質説明力強化のための制度フレームワークに関する提案(中間報告)」を公開した。

3 第三者による検証・妥当性確認の必要性

なぜ、第三者による検証・妥当性確認が必要なのか。例えば、私たちがスマートフォンを買い替えようと考えた場合、雑誌などの商品レビューで、各機種における各地点での通信速度の検証（速さ）や、専門家による機能や応答速度等の評価が良ければ、その機種が重要な選択肢の1つになることがある。自分の目で見て判断することも必要だが、すべてを自分で評価することは難しい。例えば、利用者が通信速度をいろいろな地点で計測することはコスト的にも時間的にも難しい。

図2で示すように、事業者からの説明に対して、事業者から独立しており、常に公正かつ客観的に判断を行うことが出来る第三者が専門的知見から妥当性の確認を行い、消費者に説明をすることは、消費者にとって有益であり信頼がおけるだろう。

分野は異なるが、第三者による監査制度の代表例として、会計監査がある。会計監査では、企業の会計処理の妥当性を企業とは独立した公認会計士、あるいは監査法人が第三者の立場で確認し、その結果を公開する。この仕組みが、投資家、金融機関、取引先にとって安心出来る投資、融資、取引の基本的な環境になっている。

このように、事業者の技術的主張の妥当性を、監査機関が開発技術水準と利用技術水準を考慮して第三者の立場で評価し、技術に関する専門的な知識の無い消費者にも理解出来る形で情報提供する仕組み（機器・システムの信頼性や安全性にお墨付きを与えること※4）が消費者の安心感向上に有効である。

4 ソフトウェア品質監査制度（仮称）の対象

品質マネジメントシステムに関する規格であるISO

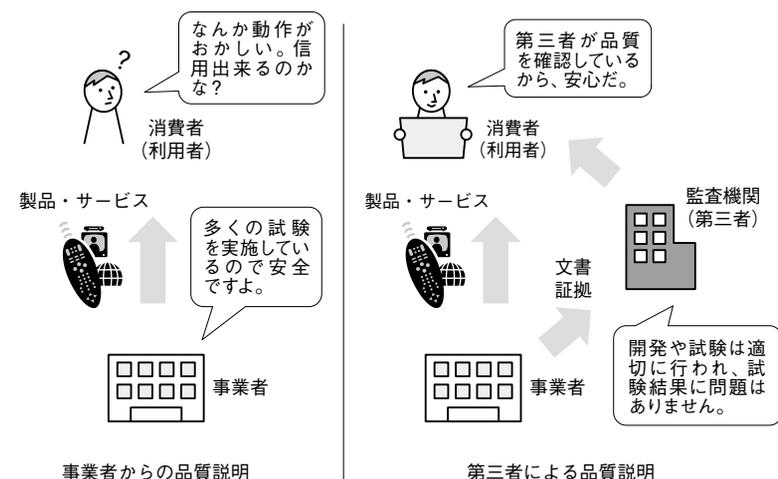



図2　第三者による品質説明（イメージ）

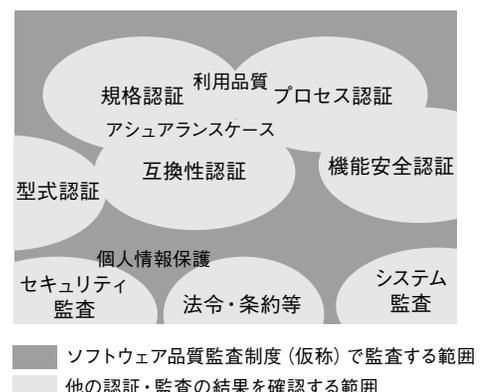



図3　ソフトウェア品質監査制度（仮称）の監査範囲

9000シリーズや環境マネジメントシステムに関する規格であるISO 14000シリーズが組織のマネジメントシステムを対象にしているのに対し、本制度の品質監査の対象は、ソフトウェアが組み込まれた製品やシステムが提供するサービス、それらが統合された統合システムそのものである。

その狙いは、消費者に対してその製品やサービスの信頼性や安全性が妥当であることを分かりやすく示すことによって、技術の専門家ではない消費者の安心感を向上させることにある。

ただし、製品やサービスの信頼性や安全性は、組織のマネジメントシステムと密接にかかわっていると考えられることから、品質マネジメントシステムなどの規格認証を取得していれば、それらに関係する審査項目については満たしているとみなすといった工夫を制度に盛り込んでいる。

つまり、図3のように規格認証やプロセス認証、セキュリティ監査等既存の制度でカバー出来る範囲（薄いグレーの部分）については、その結果の確認だけを行い、カバー出来ない範囲（濃いグレーの部分）については、本制度で定めた審査基準により、機器・システムの安全性や妥当性を確認することになる。

5 ソフトウェア品質監査制度（仮称）に対する要望

制度を有効に機能させるためには、消費者や産業界の理解が必要である。本制度設計にあたっては、消費者や産業界からの要望が検討された。その内容とそれらに対する対応方針を以下に示す。

①消費者からの要望
- 製品やサービスが安心して使えるものであることが専門家でなくても分かる制度であること
- 製品やサービスが安全に設計されていることが分かる制度であること
- 消費者の利用情報が反映される制度であること
- 障害に対して適切に対応される制度であること

②産業界からの要望
- 国際的に認知される制度であること
- 先端開発にも適応出来る高い機密保持性を持てる制度であること
- 要求される品質説明力とコストとのバランスが取れる制度であること
- 義務ではなく任意の制度であること
- 既存の規格認証との検査作業等の重複が少ない / しない制度であること
- 複数の業界を跨ぐシステムに関係する全業界間で共有・支持出来る制度であること
- 認証取得だけでなく品質向上にも有効な制度であること

③要望事項への対応方針
- 第三者による監査結果を消費者に分かりやすく伝える枠組みとする
- 消費者の利用情報や製品、サービスの障害情報を各基準に反映する
- ISO国際認証（ISO/IEC 17011※5等）の枠組みとの整合性を確保する
- 国際的に認知されている他の仕組み（宇宙分野のIV&V※6、会計監査制度等）を参考にする
- 機密保持性を持つ他の仕組み（米連邦航空局製品認証のDER※7制度等）を参考にする
- 要求される品質説明力に応じてレベル分けし、レベルごとに監査コストも考慮して監査範囲を定める
- 消費者等に対して監査の有無や監査結果を分かりやすく告知するための統一的な表示方法を定める
- ソフトウェア品質監査と同一の規格認証の審査項目については、その結果を監査で利用する

脚注

※3　検証・妥当性確認の結果、疑惑を証明する証拠は見つからなかった。

※4　類似の仕組みの例として、SGマーク（エスジーマーク）がある。SGマークは、一定の日用品の安全性を保証するための制度で、対象製品ごとに基準が定められており、基準適合品に対しては第三者機関である財団法人製品安全協会が認証し、経済産業大臣が承認し、SGマークを付与することが出来る。

※5　ISO/IEC 17011：Conformity assessment – General requirements for accreditation bodies accrediting conformity assessment bodies, 適合性評価機関を審査・認定する認定機関に対する一般要求事項を規定。

※6　IV&V：Independent Verification and Validation, 発注元やメーカ等の開発組織に対し、組織的・予算的・技術的に独立した立場で、ソフトウェア開発におけるリスクを低減し、品質を向上させる活動。

※7　DER：Designated Engineering Representative, FAA（Federal Aviation Administration）が航空規制の手続きの認証作業を委任する幾つかの指名代理人の一つ。

・規格認証と同一のソフトウェア品質監査の審査項目については、規格認証でその結果を利用出来るようにする
・異なる製品・産業分野でも同等の監査精度を担保出来るようにする
・製品・産業分野で異なる分野依存部と共通な分野非依存部を別けて審査基準を定義出来るようにする
・分野依存部については業界団体等が審査基準を策定する
・分野非依存部については公的機関が審査基準を策定する

6 ソフトウェア品質監査制度（仮称）の狙いと効果

本制度の狙いと効果を図4に示す。

6.1 利用者の安心感の向上

これまで説明してきたとおり、対象とする製品やサービスの専門家である第三者が製品やサービスの品質にかかわる事業者の主張の妥当性を確認し、分かりやすく利用者に伝えることにより、利用者の安心感の向上を図る。

6.2 国際競争力の維持・強化

製品、サービスのグローバル展開が進む中において、本制度が国内だけで通用するものは、それぞれの地域・国で定められた品質に関する規格や仕組みにも対応しなければならず、事業者にとっては大きな負担となる。その結果として、消費者が製品やサービスを受ける際の価格にも影響を及ぼしてしまう。

製品やサービスの品質に対する正当な評価を行う仕組みを確立することにより、国際的にも通用する制度にし、消費者が安全な製品やサービスを適正な価格で享受出来ると共に、国内産業の国際競争力の維持・強化を図る。

6.3 国民生活の快適性、利便性の向上と新成長分野における国際優位性の確保

今後実現が期待されているシステムとして社会レベルでの環境調和を実現するスマートコミュニティが注目されている。スマートコミュニティとは、太陽光や風力など再生可能エネルギーを最大限活用し、一方で、エネルギーの消費を最小限に抑えていく社会が必要であるとの考えから生まれたシステムであり、住宅やビル、交通システムをITネットワークでつなぎ、地域でエネルギーを効率的に授受して有効活用する次世代の社会システムのことである。

このような、品質文化が異なる産業界を横断する統合システムに対して、品質の見える化の仕組みを作ることにより、潜在的なリスクを低減し開発加速を図り、スマートコミュニティのような新成長分野における国内産業の国際優位性を確保する。また、品質の良いサービスが提供出来ることになり、国民生活の快適性や利便性の向上を図ることが出来る。

6.4 製品・システムの本質的な品質向上につながる制度

そして、何よりも第三者が製品やサービスの品質にかかわる事業者の主張の妥当性を確認する仕組みを導入することにより、本質的な品質向上が図れ、国民生活の安全性を確保する。

7 ソフトウェア品質監査制度（仮称）のフレームワーク案

これまで述べてきた消費者や産業界からの要望に対する対応方針や制度の狙いを踏まえ、中間報告で提案したソフトウェア品質監査制度（仮称）のフレームワーク案の全体像を図5に示す。

ソフトウェア品質監査制度（仮称）の狙い	ソフトウェア品質監査制度（仮称）の効果
企業の製品・システムに関する利用者や市場への品質説明力の強化	技術の専門家ではない **利用者の安心感の向上**
国際市場における日本製品・システムの品質に対する正当な評価の確立	我が国産業の**国際競争力の維持・強化**
産業界の枠を超えた品質の見える化による複数の産業界を跨り構成される高度なシステムの開発加速　（例：スマートコミュニティシステムなど）	**国民生活の快適性・利便性の向上** 新成長戦略分野における我が国産業の **国際優位性の確保**
製品・システムの本質的な品質向上	**国民生活の安全性の確保**

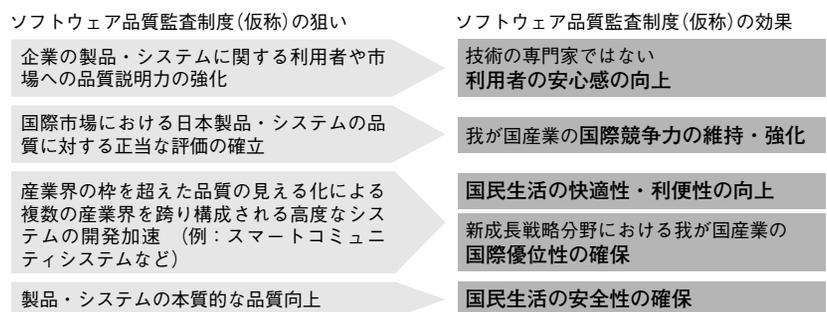



図4　ソフトウェア品質監査制度（仮称）の狙いと効果

7.1　監査の概要

監査の概要は、次のとおりである。事業者が利用者に提供する製品・サービスに対して、政府が定めた監査基準及び民間が定めて政府が認定した審査基準を参照して監査機関が監査を行う。監査にあたっては、必要に応じて政府が認定した独立検証機関が専門的な検証をして、その結果を監査機関に報告する。監査結果は公開され、利用者が活用する。

政府と民間は、本制度を運用するにあたって、利用者の利用情報、障害情報を収集、参照して各基準等にフィードバックし、更新していく。

7.2　各機関の役割

産業界横断の制度を実現するために、政府は産業分野に依存しない共通な基準を策定する。一方、産業界特有の専門性が必要な分野依存の基準の策定や、監査業務は民間で実施する（表1、表2）。

7.3　競争領域の機密保持の考え方

本制度の特長の一つは、監査を受ける製品やサービスを提供する事業者内に公認審査官（以下、内部審査官）を置くことが出来、事業者とは独立の公認審査官（以下、外部審査官）と協調して監査を実施出来る点である。

これは、外部（他社）には公開したくない競争領域の

表1　政府が担う役割

機関	役割	内容
認定機関	監査基準、監査実務ガイドラインの策定	公認審査官が遵守すべき心得や行動基準を含む監査基準及び監査実務のためのガイドラインを策定・維持する。
	認定基準の策定と認定業務	公認審査官、監査機関、審査基準策定機関、独立検証機関、審査基準策定機関が策定した審査基準を認定する。各機関、基準の認定基準を策定・維持する。
	審査基準策定指針の策定	産業分野で審査基準を策定する際の指針を策定・維持する。

表2　民間が担う役割

機関	役割	内容
審査基準策定機関	審査基準の策定	政府の認定を受けた審査基準策定機関は、監査を実施する基準となる審査基準を、産業分野あるいは製品分野ごとに策定・維持する。
監査機関	監査の実施と報告	政府の認定を受けた監査機関（監査に対応する目的で複数人の公認審査官により組織的に監査業務を実施する能力があると公式に認定された組織）あるいは公認審査官は、審査基準に従って製品やサービスの監査を行い、その結果を利用者に分かりやすく提示する。
独立検証機関	専門的な検証サービスを提供	政府の認定を受けた独立検証機関は、監査機関あるいは公認審査官の指示で監査に必要な検証作業を実施する。
公認審査官協会	監査の品質確保、公認審査官の能力の向上	公認審査官協会は、公認審査官による審査業務の査察、公認審査官の能力を維持・向上するための教育研修を実施する。

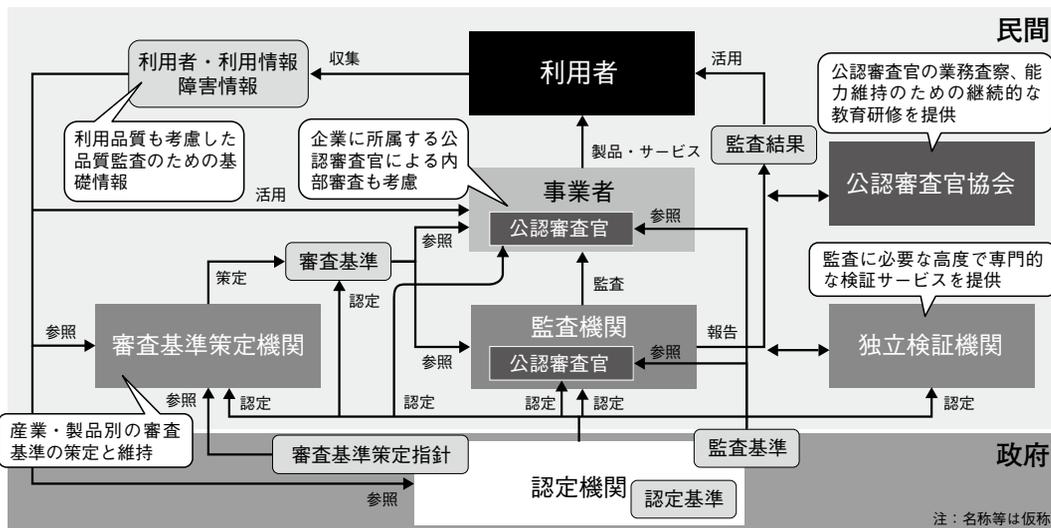



図5　ソフトウェア品質監査制度（仮称）のフレームワーク案

監査は、内部審査官が内部監査を実施し、その結果を外部審査官に報告することによって競争領域の情報の機密が保持出来るようにしたものである。更に事業者の中に内部審査官を置くことで、事業者内での品質改善が促進される効果も期待出来る。

外部審査官と内部審査官の役割を図 6 に示す。

監査計画の立案にあたっては、内部審査官が審査対象組織と協議し、開発情報の機密性を評価し、内部監査での実施と外部監査での実施に峻別して、監査計画案を策定する。監査計画案は外部審査官の合意を得て監査計画を確定する。

監査の実施にあたっては、内部監査は、内部審査官と審査対象組織で実施されるが、監査の実施状況については外部審査官に報告し、外部審査官が確認する。外部監査は、外部審査官と審査対象組織で実施されるが、内部審査官は外部監査を支援することが出来る。

監査結果の表明は外部審査官が行うが、内部審査官に意見を求めることが出来る。

7.4　流用開発、市販ソフト、オープンソースソフト活用システムの監査

ソフトウェア開発においては、新規開発だけでなく、

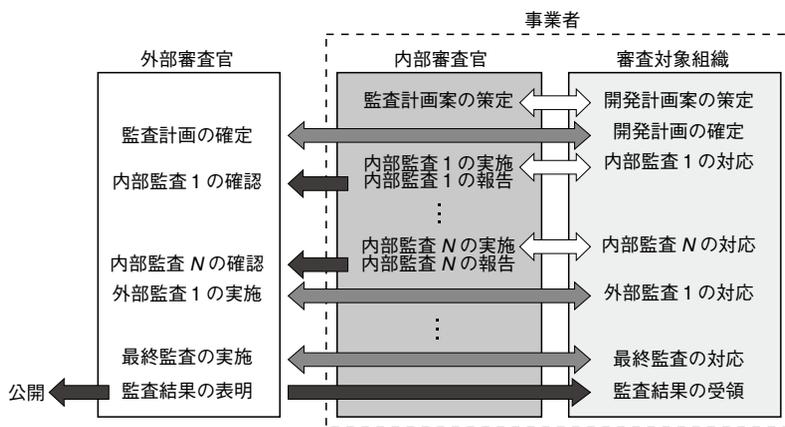



図6　外部審査官と内部審査官の役割

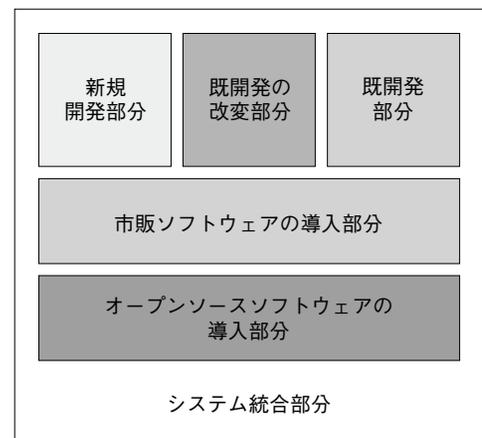



図7　流用開発･市販ソフト･オープンソースソフトを活用したソフトウェア構造例

(a) 利用者・国民影響レベル

レベル	影響の範囲・程度
4	利用者並びに利用者以外への重大な影響（代替手段による影響軽減が困難な影響） 国民への広範囲で重大な影響
3	利用者への重大な影響に加え､利用者以外への軽微な影響(代替手段による影響軽減が容易な影響)
2	利用者に限定された重大な影響
1	利用者に限定された軽微な影響
0	影響は無い／ほとんど影響は無い

(b) 産業・経済影響レベル

レベル	影響の範囲
4	国内産業への広範囲な影響
3	製品・サービスにかかわる産業に限定された影響 製品・サービスにかかわる企業以外の同一・類似産業への影響
2	製品・サービスにかかわる企業に限定された影響 製品・サービス事業以外の他事業への影響
1	製品・サービス事業に限定された影響
0	影響は無い／ほとんど影響は無い

(c) 監査レベル

産業・経済影響レベル ＼ 利用者・国民影響レベル	0	1	2	3	4
4	4	4	4	4	4
3	3	3	3	3	4
2	2	2	2	3	4
1	1	1	2	3	4
0	0	1	2	3	4

(d) 監査レベルに対応した監査内容

監査レベル	監査する審査項目	監査方法	独立検証
4	全項目	網羅監査(全件監査)	必須
3	重要項目	網羅監査(全件監査)	必須
	その他の全項目	抜取監査(サンプル監査)	任意
2	全項目	抜取監査(サンプル監査)	任意
1	重要項目	抜取監査(サンプル監査)	任意
0	対象外	対象外	対象外

図8　監査レベルの設定と監査レベルに応じた監査内容

既に開発済みのソフトウェアをベースに必要に応じて機能の追加を行う等の改変や、ソフトウェア部品として流通しているものを用いて構成される場合が多い（図7）。

新規開発の場合は、監査レベルに応じた通常の監査を実施することになるが、改変無しの既開発部分が監査済みの場合は、どのように扱うか、改変した場合はどうするか、また、市販ソフトやオープンソースソフトの部分をどのように扱うかは、現実的に監査が実施可能かどうかを含めて検討を進める必要がある。

7.5　監査レベルの設定とレベルに応じた監査内容

本制度においては、監査の程度を監査レベルとして設定することとし、利用者・国民及び産業・経済に与える影響度合いに応じて設定した。利用者・国民への影響度合いとは、製品・サービスの障害により利用者や国民が被る生命・身体にかかわる危害の度合い等である。また、産業・経済への影響度合いとは、製品・サービスの障害により産業界や国が被る経済的損失である。影響度合いが大きいものは、監査の実施においても詳細な監査をすることによって、製品やサービスの安全性、信頼性を確実に担保する。逆に影響度合いの小さいものについては、監査のコストを軽減する。利用者・国民への影響度と産業界・経済への影響度によりレベル分け（監査レベル）し、監査レベルごとに監査内容を定義した内容を図8に示す。

監査する項目としては、審査項目の中に重要項目を設け、すべての項目を対象にするか、重要項目だけにするかを判断する。また、監査方法としては、網羅的に監査するか、抜き取り監査にするかの判断を行う。特に監査レベルが高い製品、サービスについては独立検証機関による検証を行う。

7.6　利用者情報、障害情報の活用

本制度においては、利用品質（有効性、効率性、満足度）も監査対象にしている。これは、製品やサービスが複雑化すると同時に利用者も多様化しており、利用者が求めている機能やサービスとギャップが生じていないかをチェックしていくことが重要だからである。また、製品やサービスの障害情報も製品やサービスの品質向上に活用していくことが重要である。

このように、利用者情報や障害情報を、いかに収集して審査基準等に組み込んでいくか検討する必要がある。

8　おわりに

本稿で紹介したソフトウェア品質監査制度（仮称）はソフトウェアの信頼性や安全性を消費者に分かりやすく伝え、安全、安心、快適な国民生活の確保に加え、スマートコミュニティのような新成長分野における国内産業の国際優位性の確保につながるものである。

紹介した内容は枠組みの提案段階であり、現在、2013年度の運用を目指し、具体的な検討を行っているところである。したがって、今後具体化する中で、必要に応じて修正を加えていく予定である。

なお、「ソフトウェアの品質説明力強化のための制度フレームワークに関する提案（中間報告）」[SEC HP] は、下記URLから入手可能であり、詳細はそちらを参照されたい。

http://sec.ipa.go.jp/reports/20110930.html

参考文献

[SEC journal 26]　松田, 金沢：情報システムの障害状況　2010年データ, SEC journal No.26, Vol.7, No.3, pp.102-104, 2011

[SEC HP] ソフトウェアの品質説明力強化のための制度フレームワークに関する提案（中間報告）, 平成23年9月30日, SECウェブサイトよりPDFダウンロード可能, http://sec.ipa.go.jp/reports/20110930.html

[SEC journal 25] SEC journal No.25, Vol.7, No.2, pp.56-58, 2011

消費者機械安全性・信頼性保証の国際標準化

SEC リサーチフェロー
トヨタ自動車株式会社 東富士研究所
第２パワートレイン先行開発部
大畠　明

独立行政法人産業技術総合研究所 産学官連携推進部門
関西産学官連携センター
田口　研治

東京大学情報基盤センター
スーパーコンピューティング研究部門
松野　裕

独立行政法人産業技術総合研究所 知能システム研究部門
ディペンダブルシステム研究グループ
中坊　嘉宏

消費者機械とは、一般ユーザが利用する自動車、家電、サービスロボットなどの機械製品に対する造語である。これらは、産業機械とは異なり、技術者の手を離れ、多様な環境で多くのユーザに利用される。些細な不具合が深刻な事態を招く可能性があり、安全性と信頼性が特に重要である。ここでは、消費者機械の安全性・信頼性・セキュリティを含むディペンダビリティを保証する組込み制御ソフトウェア開発に関する国際標準化へのIPA/SECでの取り組みを紹介する。

1 はじめに

消費者機械は一般のユーザが利用する自動車、サービスロボット、家電、スマートハウスなどの機械製品に対する造語である。消費者機械は表１に示すように産業機械とは大きく異なる。しかしながら、現在の安全性に関する標準化の枠組みはISO 12100、ISO 14121やIEC 61508のような産業機械や工業プラントに対する安全の枠組みを拡大利用している。消費者機械安全に対する標準化の枠組みは不十分であり、自動車やサービスロボットなどの領域で電子制御システムの機能安全規格[ISO 26262]が個別に制定されている。

表1　消費者機械と産業機械の違い

	産業機械	消費者機械
生産数	a few ↔ many	a huge number
ユーザ	experts	general users
要求コスト	high	sufficiently low
メンテナンス	現場 (strongly managed)	ユーザ、 サービスステーション (weekly managed)
環境	工場環境 (ほぼ定常)	生産現場
		ユーザ環境 (open, dynamic and diverse)

表１の中で最も重要な特徴は、消費者機械は技術者の手を離れて、多くのユーザに多様な環境で長期間利用されることである。状況によっては、些細な不具合が重大な問題につながる可能性があり、安全性と信頼性は特に重要である。更に、開発中に悪意のあるコードが挿入される可能性を否定出来ない時代が来ると予想され、情報ネットワークを通じて問題が深刻化する事態に対応するセキュリティ保証が重要となりつつある。

典型的な消費者機械である自動車は高い安全性と信頼性が求められるが、同時にクリーン化と気候変動への対応が求められ、自動車の制御システムは急速に複雑化している。自動車のエンジン制御だけでも、図１に示すように、筒内空気量推定、燃料噴射制御、点火時期制御、エンジントルク制御、排気ガスエミッション低減、異常

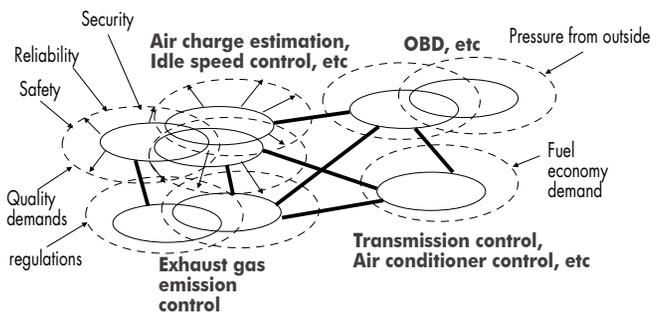



図1　複雑システムとしてのエンジン制御

図2　社会システムとの類似性

図3　物理システムとソフトウェアの頻繁な相互作用

図4　ソフトウェア開発者が想定する開発プロセス

診断などのシステムが複雑に絡み合っている。それぞれのシステムが各種の要求で独自に成長していくので、製品の世代を越えて信頼性を保証し続けることは容易ではない。更に、自動車は単体としての機能や効率向上を図ることは当然として、交通、配送、エネルギー供給、情報システムなどと連携して付加価値を創造する時代になっており、自動車制御システムは一層複雑化が進展すると予想される。このことは、ほとんどすべての消費者機械にあてはまる。

社会システムは図2に示すように交通、配送、エネルギー供給、情報システムなどが絡み合って複雑なネットワークを構成している。ある時点での最適化だけではなく、長期間にわたる総合的な計画が極めて重要である。この構造は、先のエンジン制御システムと全く同じである。

このような異種なシステムが組み合わされた複雑なネットワークシステムで世代を越えて管理しなければならないことは、今日の社会システムや製品開発に共通な課題である。消費者機械はそのような複雑なネットワークを構成する要素でもあり、今後、些細な不具合が波及して、重大な社会問題を引き起こす可能性を排除することは出来ない。すなわち、消費者機械の安全性、信頼性、セキュリティを含めたディペンダビリティを保証する枠組みを構築することは、今日の危急な課題である。

2　消費者機械の組込み制御ソフトウェア

消費者機械の組込み制御ソフトウェアは、図3に示すように、物理システムと数ミリ秒程度の間隔で頻繁な相互作用を持っている。このため、ソフトウェアだけでは消費者機械のディペンダビリティを保証することは出来ず、人間、制御対象や走行環境も考えなければならない。また、極めて多様な使い方を想定せねばならず、すべてのソフトウェア要求を事前に把握することは容易ではない。消費者機械のハードウェア開発でも同様であり、ハードウェアの設計要求に変更が無い開発を実現することは容易ではない。このため、自動車会社は継続的改善によって、安全で信頼性の高い車を合理的な価格で提供出来るシステムを構築してきた。

おそらく、ソフトウェア開発者が想定する開発プロセスは図4のようだろう。多くの場合、検証段階で要求とシステムの振る舞いの不一致が顕在化し、開発のやり直しになる。このため、ソフトウェア開発者は要求の精度を上げるためにクライアントの要求の背景を知ろうとする。しかしながら、実際にはこのアプローチには無理がある。すなわち、背景から潜在的要求を見出そうとしても、その分野の実験データ、経験、物理知識、数学知識が必要となり、ソフトウェア開発者の負担が非常に多い。

ソフトウェア開発要求の背景には、図5のように、実

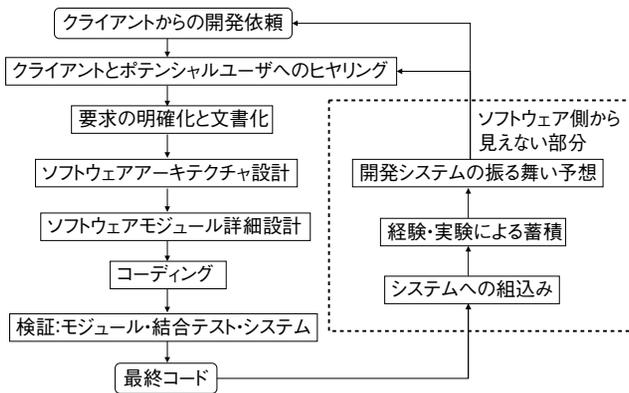



図5　システム開発の全体プロセス

験やデータの蓄積、理論解析やシミュレーションなどがある。開発には未知な要素が必ず含まれるので、システム開発者は、実際に実験してみなければ必要な要求が分からないことが多い。また、実験のためには、組込みソフトウェアが必要なので、実際の消費者機械の開発は図5のような閉ループとなる。すなわち、繰り返し開発は必須である。

3　ディペンダビリティ保証の着目点

図6はシステム制御技術者から見た制御アルゴリズム開発を示している。システム制御技術者は制約条件、制御対象の振る舞いの知識（制御対象のモデル）、望ましいシステムの振る舞い定義があれば、制御アルゴリズムを導くことが出来るが、これらを開発における想定と呼ぶことにする。

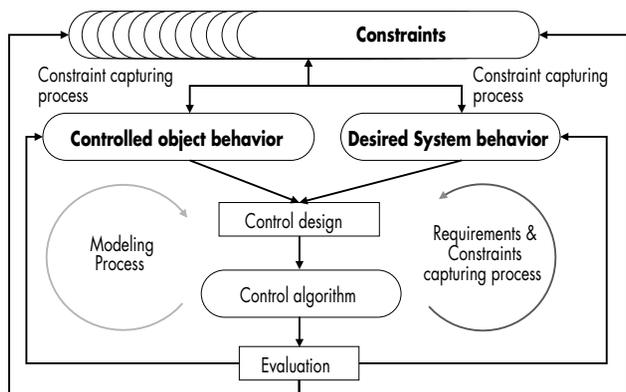



図6　システム制御技術者から見た制御アルゴリズム開発

想定に着目したプロセスを図7に示す。まず既存の知識に基づき想定を作り、開発を行う。一般に、ある部分は想定通りだが、ある部分は想定から外れている。それを知るには試してみるしかない。試行によって想定外の部分が見つかれば、それを正して新しい想定を作成する。それを繰り返して精度の高い想定を作り出す。自動車会社は、このような繰り返しと継続的改善を積み重ね、合理的な価格で信頼性の高い車を提供するシステムを構築してきた。一般に、日本の消費者機械製造会社は想定内のことは非常に良く管理している。従って、ディペンダビリティをより一層強化するためには、素早い繰り返しに基づく想定の改善が重要である。

素早い繰り返しは、対象が複雑なほど初期想定の精度が重要となる。初期想定が悪いと、次に想定のどこを修正すれば良いかという判断が難しくなり、繰り返し開発の効率が著しく低下する。精度の高い想定を得るためには、高精度なシステムの振る舞い予測が必要である。このためには、制御対象の操作量と外乱から制御量と計測量までの関係を微差分方程式で表したモデルを用いることが効果的である。このモデルの記述にはSimulink[SIMULINK HP]とStateFlow[STATEFLOW HP]が用いられることが多い。システムが時系列的にどのように変化するかを知れば、それに応じた的確な対応が出来る。このような、制御対象と制御装置の動的振る舞いモデルを利用した開発をモデルベース開発(MBD※1)と呼ぶ。ここでのモデルはソフトウェア技術者が使うモデルの定義とは大分違う。一般にソフトウェア技術者が使うモデルとは関係図を意味し、制御対象やECU※2の

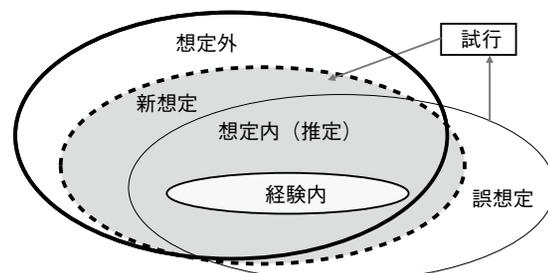



図7　想定外への対応

時系列変化を定量的に予想するものではない。しかし、両者は独立なものではなく、UML※3やSysML※4と動的振る舞いモデルとの統合の重要性が指摘されている。

制御装置の動的振る舞いモデル（制御モデル）はECUの振る舞いを正確に記述しているので、制御モデルから組込み制御用のコードを自動生成することが出来る。更に、実際の制御対象と制御モデルの組み合わせ、制御対象モデルと実ECUを組み合わせたシミュレーションも可能であり、効率的な検証環境を得ることが可能である。例えば、SILS※5、MILS※6、HILS※7とRapid Proto typing ECUを様々に組み合わせて、素早い繰り返しを加速することが出来る。

組込み制御システムのソフトウェアには、ソフトウェア開発者が意識していないとしても、制御対象の物理と数学が反映されているので、その情報を使うとソフトウェア開発効率を向上出来るのではないかという期待が湧く。例えば、自動車エンジンは連続事象システムなので、その制御も連続事象システムになる可能性が高い。しかしながら、ソフトウェアだけを見るとすべてが離散事象システムであって、どこが連続事象システムを反映しているか全く分からない。従って、本来的に離散事象システムである部分と連続事象システムを区別すれば、制御対象の特徴をうまく使えるかも知れない。システム制御理論によれば、すべての連続事象システムは（1）のように表すことが出来る。

$$\frac{dx}{dt}=f_c(x,u),\quad y=g_c(x,u) \tag{1}$$

ここで、$x \in R^n$は状態量であり、$u \in R^m$は操作量、$y \in R^p$は計測量である。シングルタスクを前提として、十分短い時間で離散化すれば、

$$\begin{aligned} x(k+1)&=f_d(x(k),u(k)),\\ y(k)&=g_d(x(k),u(k)) \end{aligned} \tag{2}$$

で表すことが出来る。（2）でkはサンプリング時刻を表す。f_dとg_dは静的な関数であるので、マップで近似することが出来る。従って、連続事象システムに対応するソフトウェアはマップとシフト演算で再構成することが出来る。実は、エンジン制御システム開発の最終工程で見つかった不具合をコードの変更無しに、マップの値を変更するだけで、解消するという芸当が出来る。上記はその理論的根拠を与え、素早い繰り返しを可能とする手段でもある。一度、（2）式を実現するコーディングを行えば、制御対象のハードウェアが変わっても再コーディングは不要という可能性が出てくる。実際には、xの次元が非常に大きくなりやすいという問題があるので、多次元のマップを低次元のマップの組み合わせやその関数補正にしなければならない。これは、f_dとg_dの数学上の近似問題と捉えることが出来る。また、そのような組み合わせで実現されたものを検証することは静的関数なので容易であり、自動コード生成も容易である。自動車エンジン制御システム開発ではモデルベース適合が進展しているが、その手法とも相性が良い。

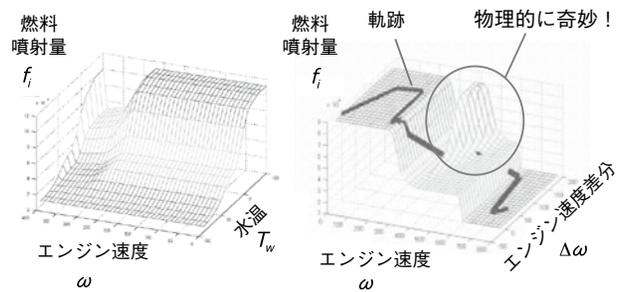



図8　状態方程式表現の適用例

エンジン始動時の燃料制御ソフトウェアを（3）のような状態方程式表現として、右辺を調べた例が図8である。

$$x(k+1)=f_d(x(k),T_w(k)) \tag{3}$$

ここで、$x(k)=[f_i(k),\omega(k)]'$、ωはエンジン速度、T_wは冷却水温度、f_iは燃料噴射量、Δは差分を表す。図8の左図は冷却水温が低くなるほど、また、エンジン速度が低くなるほど燃料噴射量が増えていることが分かる。しかし、350rpm付近に燃料噴射量の落ち込みがある。右図は物理的には予測出来ない奇妙な山が表れている。実験での時間変化を実線の軌跡で示しているが、この山

脚注

※1　MBD：Model-Based Development
※2　ECU：Electronic Control Unit
※3　UML：Unified Modeling Language
※4　SysML：Systems Modeling Language, UMLを拡張した、システムをモデリングするための記述言語。
※5　SILS：Software In the Loop Simulation
※6　MILS：Model In the Loop Simulation
※7　HILS：Hardware In the Loop Simulation

は通っていない。しかし、このコードは潜在的に問題があることを暗示している。このように、物理と数学を用いれば、新しい視点でソフトウェアを検証することが出来る。

以上を整理すると、ここでの取り組みの着眼点は下記の3点である。

(1) 素早い繰り返し

(2) 制御対象と制御装置の動的振る舞いモデル

(3) 物理と数学のソフトウェア開発利用

上記の着眼点は日本のお家芸であり、実は、極めて日本的であり、ある意味、実際に行われている開発を国際標準化に乗せる取り組みとも言える。

4 ソフトウェア開発プロセス

図9はソフトウェア開発者がよく提案する開発フローの概略を示している。要求獲得から始まり、ソフトウェア仕様書を作成し、コーディングを行い、コンパイル・リンクしてオブジェクトコードを実装し、制御定数などの調整を行い、システムとしての検証を行う。ここで、問題が発見されれば、要求を修正し、同じ工程を繰り返す。このプロセスはトップダウン的であり、欧米的ということが出来るだろう。このプロセスの問題は、繰り返しが必要な制御設計プロセスの中に時間がかかる実装プロセスが入り込んでいることである。このプロセスは実現目標になりがちだが、実際には実行が難しい。現場で使われているプロセスは図10のように、仕様書を作成する前にコーディングを行い、それを実装したシステム検証を繰り返して、コーディング品質を素早く向上させる。この過程は先行開発と呼ばれる。コードの完成度が十分になったところで、ソフトウェア仕様書を作成し、

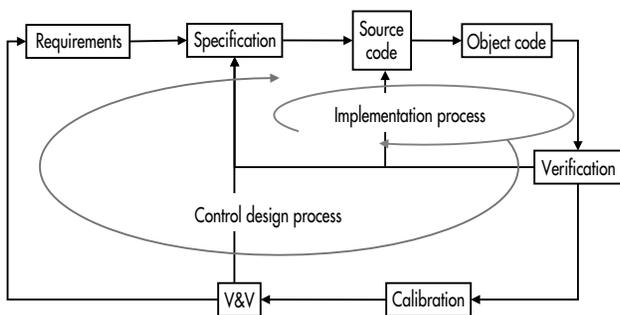



図9　ソフトウェア開発者が提案するプロセス

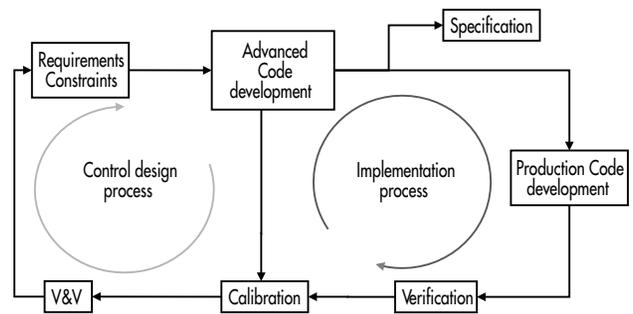



図10　実際の組込み制御ソフトウェア開発

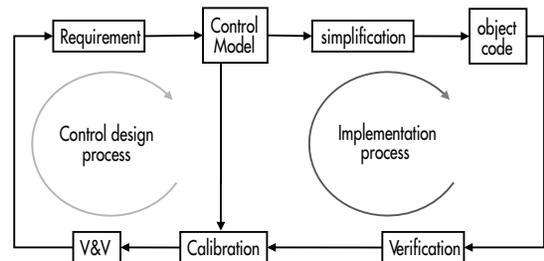



図11　モデルベース開発プロセス

ECUサプライヤにソフトウェア開発依頼をする。この過程は量産開発と呼ばれる。ここで重要なのは、ソフトウェア仕様書の前に機能確認されたコードが存在していることである。

しかし、図10はソフトウェア開発の生産性が高いとは言えないだろう。そこで、図11に示すモデルベース開発プロセスが提案されている。ここでは、Simulink/StateFlowなどを用いて実行可能な制御モデルを開発する。制御モデルからは自動コード生成によって、実装コードを得ることが出来、直ちに検証実験を行うことが出来る。素早い繰り返しで制御モデルの完成度を上げ、実装プロセスに引き渡す。実装は「制御モデルで定義された入出力関係を目標ECU上で許容誤差内の再現」と定義される。入出力関係は厳密に定義されているので、物理的意味などに拘らず、実装が行える。図11では、実装プロセスの最初に簡易化という工程を置いている。これは、制御モデルで定義された入出力関係をメモリ容量、実行時間、実装誤差の制約下で最も簡単な実現をすることを意味する。例えば、現在でも制御モデル記述に浮動小数点を用い、実装は固定小数点で行われることがある。これは、簡易化の一例である。しかし、ここでの簡易化の意味は第3章で述べた状態方程式表現のように、ロ

ジックの意味を捨てた許容誤差内での入出力の再現を意味し、従来よりも更に進化することを提案している。

5 素早い繰り返しと国際標準化

開発した消費者機械のディペンダビリティを第三者に納得してもらうためには、第3章で述べた制約、制御対象のモデル、望ましい振る舞いの定義を開発したシステムが満たしている証拠、及び、なぜそれがディペンダビリティの証拠たり得るかという論証のセットが必要であるとされている [ALGIRDAS2004] [GEORGIOUS2007] [OMG 2010] [TIM2004] [WAC2004]。このセットをDC※8と言う。素早い繰り返しを前提にするならば、DCは開発中にリファインされ、開発終了時に完成することになる。認証書類は完成したDCから生成すれば良い。ただし、開発中にDCをリファインするのは大変なので、可能な限り自動化したい。このためには、プロセスとデータ間の関係をメタモデルで形式化することが不可欠である [OMG2010]。図12に示す各矢印にautomatedの文字を入れてあるが、プロセスやデータを形式化し、モデルやデータ管理を可能な限り自動化することを目指すという意味である。この概念を入れて、図11を書き直すと図13のようになる。

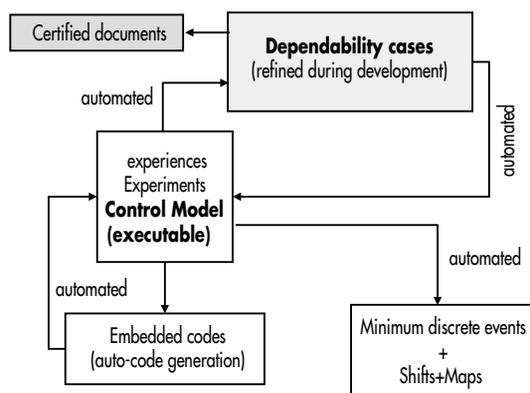
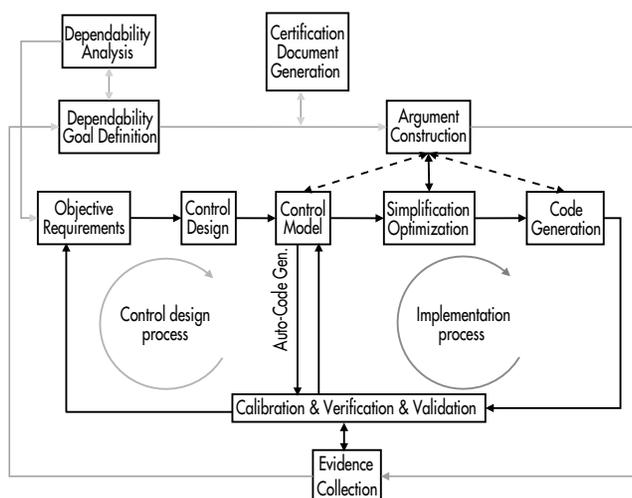



図12 ソフトウェアとDCの同時開発

図13 提案するプロセス

6 これまでの取り組み

2010年9月にボストンで開催されたOMG※9で、OMG会長兼CEOのRichard Mark Solely、電気通信大学 新 誠一教授と著者の一人である大畠が会談し、消費者機械安全に対するOMGでの標準化活動の可能性に関して議論を行った。その場で、OMGメンバに対して議論の内容を紹介して欲しいと依頼があり、翌日、急遽発表をすることになった。また、次のアクションとして日本と米国で消費者機械安全に関するworkshopを開催することが合意された。日本でのworkshopはIPA/SEC主催セミナーとなり、東日本大震災と福島第一原発事故の影響が残る中、Solely会長が来日し、4月19日に100名を超える出席者を得て開催された。日本からは新、中坊、大畠が講演を行った。続いて、ソルトレークシティで開催されたOMG会議の中で6月22日にセミナーが開催され、翌日、System Assurance Task Force（SysA）[SYSA1 HP] [SYSA2 HP]で議論を行い、9月にフロリダで開催されるOMG会議にRFI※10とホワイトペーパを提出することになった。そのときの議長は著者の一人である田口である。実は、22日の夜、Safety Assurance分野で影響力があり、ISO 26262の執筆にもかかわったYork大学教授のTim Kelley、松野、田口、中坊、大畠は夜遅くまで議論を行い、図12の概念がまとまった。ホワイトペーパの執筆にあたり、松野、田口、中坊、大畠の共著でYork大学で9月に開催されるWDSoS※11 2011用に論文を作成することにした [MATSUNO2011]。この発表は松野が行った。23日のSysAでの主な議論は、

脚注

※8 DC：Dependability cases
※9 OMG：Object Management Group
※10 RFI：Request for Information
※11 WDSoS：Workshop on Dependable System of Systems

提案が消費者機械に限らない一般的なもので、なぜ、消費者機械に限定するのかということであった。領域が広がると、まとまらない危険性が高くなるので、まずは消費者機械からとしたが、この問題は再燃する可能性が高い。9月のOMG会議のSysAで若干の修正要請があったが、RFI とホワイトペーパは基本的には発行が認められた。次回は12月にサンタクララでの開催であるが、修正案を提示して2012年の年初に発行にこぎ着けたい。発行されたRFIに対しては、誰でもホワイトペーパをOMGに投稿し、意見を述べることが出来る。読者にも広く意見を求めたい。

7 まとめ

複雑化が進む消費者機械と消費者機械がその要素として動作する社会システムにとってディペンダビリティの確保が危急の課題である。ここで提案している素早い繰り返し、動的振る舞いモデルの導入、物理と数学のソフトウェアへの導入の心は日本的開発手法そのものである。日本は合理的な価格で、高機能、高品質な消費者機械を世界の提供してきた実績がある。しかし、日本的な方法が国際標準に取り入れられることは必ずしも多くはない。それは、日本的開発手法の科学や工学の裏付けは必ずしも十分ではなく、説得力が十分無かったのかも知れない。ソフトウェアとディペンダビリティの同時開発と形式化による自動化の推進という衣をまとい、より先進的な開発手法として、世界に向けて発信し、国際標準化の枠組みに組み入れたい。ぜひ多くの方のご支援をお願いしたい。

参考文献

[ALGIRDAS2004] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr : Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Secur. Comput, 1（1）: 11-33, January 2004

[GEORGIOUS2007] Georgious Despotou : Managing the Evolution of Dependability Cases for Systems of Systems. PhD thesis, Department of Computer Science, University of York, 2007

[GSN2010] GSN contributors. DRAFT GSN standard version 1.0, 2010

[ISO 26262] ISO 26262 road vehicle - functional safety -, part 1 to part 10. Technical report, 2010

[MATSUNO2011] Matsuno, Taguchi et al. Iterative and Simultaneous Development of Embedded Control Software and Dependability Cases for Consumer Devices, WDSoS11

[OMG2010] OMG, Argument metamodel (ARM), OMG Document Number Sysa/10-03-15

[RAILTRACK2000] Railtrack. Yellow book 3,Engineering Safety Management Issue3, Vol. 1, Vol. 2, 2000

[SIMULINK HP] http://www.mathworks.co.jp/products/simulink/

[STATEFLOW HP] http://www.mathworks.co.jp/products/stateflow/

[SYSA1 HP] http://www.omg.org/news/meetings/tc/agendas/ut/SysA_info_day.htm

[SYSA2 HP] http://sysa.omg.org/

[TIM2004] Tim Kelley and Rob Weaver. The goal structuring notation - a safety argument notation. In Proc. of the Dependable Systems and Networks 2004, Workshop on Assurance Cases, 2004

[WAC2004] Workshop on Assurance Cases : Best Practices, Possible Obstacles, and Future Opportunities, DSN 2004, 2004

形式手法導入のための産学連携PBLの活用

形式手法人材育成WGリーダー
大学法人 九州大学大学院システム情報科学研究院／システム情報科学府　教授　工学博士
荒木　啓二郎

近年、我が国においても形式手法に基づくソフトウェア開発が注目を集めているが、その導入にはまだ障壁が高いと思われているためか、ソフトウェア開発の現場への普及が進んでいるとは言い難い状況である。九州大学では、大学院修士課程におけるPBL※1科目において、企業との連携のもとに、企業での実際のソフトウェア開発プロジェクトの一部で学生チームが形式手法を適用した結果、形式手法に対する企業側の理解が深まるとともに、導入に関する具体的な見通しを得ることが出来た。本稿では、我々の経験に基づいて、形式手法導入の一つの有効な方法としてのPBL実施事例を述べる。

1 はじめに

近年、我が国においてもシステムの信頼性や機能安全に関連してフォーマルメソッド（Formal Methods）に対する関心が高まっている。しかしながら、大学や公的研究所や民間企業において、形式手法に関する研究開発や適用の実践は、まだ限られた組織で行われているに留まっている。本稿では、フォーマルメソッドに関心があるものの実際のソフトウェア開発プロジェクトへの導入までには至っていない企業と、大学との協同のもとに、大学院修士課程におけるPBL科目として学生チームがフォーマルメソッドの導入を試行した事例を述べる。

図1　九州大学の社会情報システム工学コースにおけるPBL科目

2 PBL科目の概要

九州大学大学院システム情報科学府情報知能工学専攻社会情報システム工学コースでは、先導的なIT技術者及び研究者の育成を目指して、大学院修士課程において産学連携のもとに実践的な教育を行っている。2007年度から開講して、年を重ねるとともに産学連携の成果が現れてきつつある。この教育コースでは、企業の第一線で活躍している技術者研究者並びに経営者による講義やワークショップの他に、修士課程2年間のうちに3回にわたるPBLや1カ月から2カ月程度の比較的長期のインターンシップなどを実施している（図1）。

PBLでは、学生が5名程度のチームを構成して、各チームでソフトウェア開発に取り組む。開発対象には、社内利用ツールの開発や製品の一部の試作など、企業における実際の開発あるいはそれに近いものも含まれている。学生達は、PBL科目の履修を通して、ソフトウェア開発という知的な集団活動にかかわる各種の問題を体験し、それらを解決する基本的な素養や知識を身に付ける。

脚注

※1　PBL：Project Based Learning

九州大学の本コースでは、上述のように修士課程2年間で3回のPBL科目を履修する。本コース開講当初は、PBL第二とPBL第三とでチーム編成も変更して異なる開発プロジェクトを実施していた。これは、出来るだけ多くの開発対象や役割分担を経験させることを意図したためであったが、1学期で完了させる開発プロジェクトでは期間が短すぎて習得出来ることが限られているという反省のもとに、現在では図1 に示すように、PBL第二と PBL第三とで一貫性をもたせるテーマ設定としている。必ずしも同一の対象システムの開発を2学期で継続して行うとは限らないが、PBL第二と第三とを連携させて1年間かけて実施することにより、問題領域や開発手法などに関して、より理解と経験を深めることに効果が上がっている。

3 PBLを利用した形式手法適用事例

上述の PBLにおいて、産学連携により、形式手法の適用を伴うシステム開発のテーマを実施した。2010年度後期には、通信制御システム開発環境の開発を対象として、上流工程での品質向上策を提案するというテーマを実施した [ARAKI2011][YAMADA2011] 。続く 2011年前期では、提案策の有効性を実証することを目的として、企業内webシステムの一部の開発を行った[SHINOZAWA2011]。

いずれのプロジェクトも、図2に示す体制で実施した。基本は、企業の技術者が顧客役を務め、学生チームが顧客からの要求を満足すべく開発作業を進めるというものである。形式手法を専門とする教員がスーパバイザを務めた。一方では、チームによるシステム開発の実践を通してプロジェクトマネジメントに関する知識と経験を習得するという学習上の目標もあるので、PBL 担当教員がPMO※2を務めて、プロジェクトの進捗を管理した。

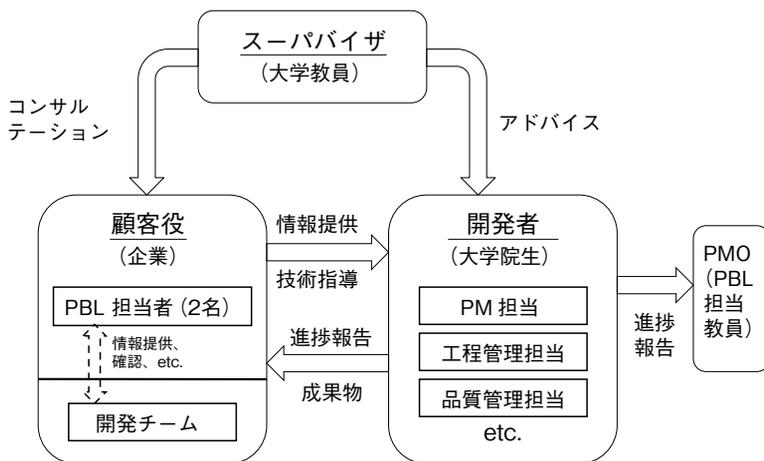
図2　PBL の実施体制

PBL期間中、企業の技術者が毎週一度顧客役として進捗の報告を受け、各種生産物のチェックと認証を行った。その際に学生達は、技術者からドメインに関する知識や提出する各種文書に関する具体的な指導を受けて、プロジェクトの進め方や解決すべき課題の認識や議事録の書き方、そして成果物のまとめ方などを学んだ。産学連携のPBLを通じて、学生達は企業の現役技術者から、システム開発に関する種々の知識や技術を学ぶ貴重な機会を得ることが出来た。

スーパバイザ役の教員は、この学生チームと顧客との会議に出来る限り参加して、プロジェクトの進行を見守った。顧客役の企業とは、適宜、プロジェクトの進め方や落としどころなどについて意見交換や打ち合わせを行った。学生チームに対しては、学生自身の活動に基づく自らの気付きや自発的学習を促進させることを意図して、どうしてもここで一言述べておかねばならないと我慢出来なくなったときに限り発言するにとどめるよう心がけた。

3.1　PBL第二：仕様の品質向上

本PBL第二は、上述のように2010年度後期に実施された。修士1年の学生5名からなるチームに対して、企業が顧客役として、システム開発の上流工程においてシステムの品質向上を図る方法を提案することを要求した。学生チームは、仕様に着目して、仕様の品質を向上させることがシステム自体の品質向上につながると考えて、形式仕様記述言語VDM++※3を用いて仕様を厳密に記述することにした。

学生達は当初、開発対象システムのドメインに関する知識は持っておらず、また、フォーマルメソッドの知識や経験もほとんど有していなかった。成果物に対する具体的なイメージも持たず、また、どのようにしてプロジェ

クトを進めれば良いかも分からぬまま、文字通り手探り状態で開始した。そのためということもあって、本PBLは、図3 に示すように、計画、調査、検討、提案の4つのフェーズを経て実施された。

最初の計画フェーズで、まず開発対象のドメインの学習とVDM※4の習得を行いながら、PBLの計画を練った。この計画フェーズにおいて、計画以降の3つのフェーズを順次実行することで本プロジェクトを遂行することと定めた。各フェーズの終わりには、それまでの進捗を評価し、その後の計画の見直しを行った。授業科目であるPBL 第二としての開発作業期間終了時には一応の成果物を提出したが、もう少し考察を加えて成果を整理したいとの学生チームの希望により、1週間あまりの追加作業を行った。図3 の右下にある「発展」フェーズがそれである。

このPBL 第二で作成して、顧客役の企業に「納入」した成果物を表1 に示す。品質向上手法提案書及び品質向上手法説明書が、顧客の要望に対する直接的な納入物である。それらを作成するために具体的な開発対象システムの要求仕様書に基づいて VDM によるシステム記述を行った際に得られた VDM 記述（VDM記述、VDM記述説明書）とシステム要求仕様書の問題点の指摘とそれに対する回答（分析シート）は、開発対象ドメインに対するより深い認識と理解を得るための貴重な資料となる。また、日本語で記述された従来の開発文書の問題点及び改善の方向を示唆するものである。更に、形式手法の初学者としての学生が VDM を学び習得する過程で、その経験や知見を取りまとめたVDM 知見集は、今回このPBL を実施した企業に限らず、これから形式手法の学習を行おうという人達に有益である。

今回の PBL 第二では、学生にとって知識や具体的経験があまりないソフトウェア開発プロセス、しかもその中の上流工程を対象にして、品質向上というこれまた学生には馴染みの少ない目標に取り組んで貰った。上述のように、対象もよく分からない、進め方も分からないという五里霧中の状況で、学生達は短い期間でよくやったという

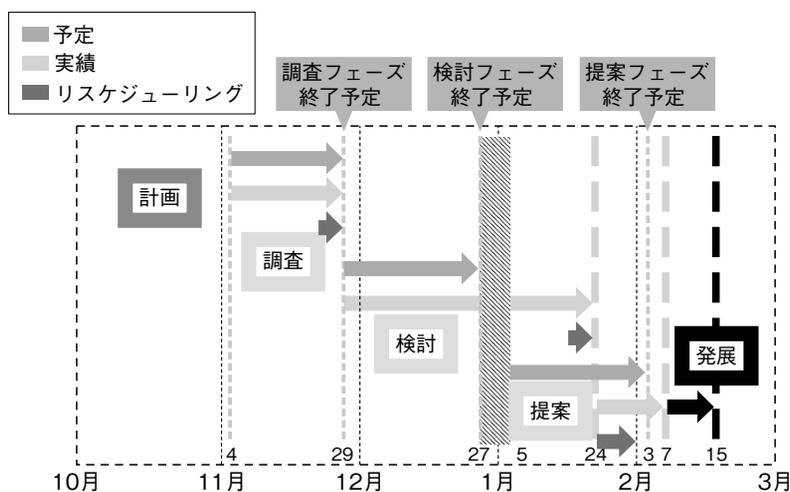



図3　PBL 第二の実施経過

表1　PBL 第二における成果物

工程	成果物	分量	説明
計画	プロジェクト計画書	Word 25枚	対象システム品質向上手法提案プロジェクトにおいて、プロジェクト管理の規約と手順を記したもの
計画	品質管理計画書	Word 22枚	対象システム品質向上手法提案プロジェクトの成果物の品質確保を行う手順を示したもの
調査	VDM知見集	PPT 42枚	VDMを経験したことのない大学院生がVDMを導入する際に得られた知見を編纂したもの
調査	ドメイン調査資料	PPT 44枚	班員で分担して行う各ドメインに関する調査結果を班内で共有するために用いた資料
検討	VDM記述	仕様部：497step テスト部：1,706step	システム仕様書における仕様検査対象範囲に対するVDM記述
検討	VDM記述説明書	Word 20枚	上記VDM記述について説明した文書
検討	分析シート	29項目	システム要求仕様書において、VDM記述を行う際に不足している情報を分析した記録
提案	品質向上手法提案書	Word 4枚	本プロジェクトにおいて提案する手法を記載した資料
提案	品質向上手法説明書	Word 36枚	上記の提案書の詳細を補足する文書

脚注

※2　PMO：Project Management Office
※3　VDM++：ISO で標準化された汎用的な仕様記述言語 VDM-SL（Specification Language）に対して、主にオブジェクト指向の拡張を行った言語。
※4　VDM：Vienna Development Method, IBM のウィーン研究所で開発された形式手法の1つ。

のが素直な感想である。顧客役を務めた企業担当者からも、成果物や学生の成長ぶりを高く評価していただいた。

学生自身も、チームでプロジェクトを遂行することの難しさ、困難を乗り越えたり問題を解決したりすることの喜びを経験して、達成感を感じた。それでも、他のPBLチームでよく見られる動くモノを実現することなく、上流工程における形式仕様記述と手法の提案に終わったことに対する学生達の欲求不満から、次学期のPBL第三では、実際に動くシステムの実現までを範囲とすることとした。

3.2　PBL第三：形式仕様記述を取り入れたシステム開発

前述のPBL第二では、学生達は上流工程の重要性を理解はしたものの、仕様の品質向上策の提案に留まったことによる不完全燃焼感があった。そこでの提案が有効かどうかを実証するため、加えて、上述のように動くシステムの実現まで行いたいという欲求をも満たすために、PBL第三では形式手法を利用した高品質なWebシステムの構築という課題に取り組んだ。その実施経過を図4に示す。要求定義フェーズ終了時に、計画の見直しを行った。実施体制は、PBL第二と同様に図2に示す通りである。ただし、学生チームは、PBL第二のときの5名から1名減って4名となった。

PBL第三における開発では一般的なウォーターフォールモデルに基づいて開発を行った。開発過程を図5に示す。形式手法の適用による高品質なシステムの開発という顧客からの要望を満たすため、ウォーターフォールモデルにおける要求定義フェーズにおいて、PBL第二で提案した形式手法の導入法に従って、VDMを用いた開発対象システムのモデル化とVDM Toolsのインタープリタを利用した仕様の実行評価による妥当性の確認を行った。

図4　PBL第三の実施経過

図5　PBL第三における対象システム開発過程

まず、顧客から提供された要求仕様書を基に、要求定義フェーズで作成される仕様の品質を向上させるために、VDMを用いたシステムのモデル化とその検証を行った。このフェーズでは、図5に示すように、要求仕様書を基にして、必要な機能の概念レベルでの機能フロー図を作成し、次に、要求仕様書と作成した機能フロー図とを基に抽象的なクラス図を作成した。次に、作成したクラス図を基にVDM++を用いて、事前条件と事後条件からなる陰仕様記述を行った。これに対して構文検査と型検査を行った後、段階的に詳細化を行って仕様の実行評価が可能な陽仕様記述を作成した。ツールを用いてVDM++の実行評価という仕様のレベルでのテストによって、システムモデルの妥当性の確認を行った。

続く設計フェーズでは、要求定義フェーズで作成したVDM陽仕様記述から、プログラミング言語での実現へ向けての詳細化を行った。それと同時に、仕様記述として書き表さなかった部分の詳細化を行った。本プロジェクトでは、作成したVDM陽仕様記述と抽象的なクラス図を基に、実際の

プログラミング言語に即したクラス図を作成した。併せて、エラー処理と画面に関する設計を行った。

実装フェーズでは、上述の設計に基づいて PHP を用いてソースコードを作成した。

テストフェーズでは、実装フェーズで作成したソースコードに対して、単体テスト、結合テスト、システムテストの3種類のテストを行った。ここでは、要求定義フェーズで行った仕様の実行評価の際に使用したテストケースを流用したテストケースを実行することで、仕様を満足するソースコードが作成されたことを確認した。

今回のプロジェクトにおける開発の各工程における開発工数の比率を表2 に示す。要求定義に 38% の工数をかけており、これは、例えば [IPA/SEC-1] で報告された開発プロセスにおける 9.8% と比較すると4倍近い。その分、開発対象システムに関する分析や確認を上流工程において前倒しで行うことによって、上流工程において品質の確保を行うとともに、下流工程、特に、テストの負担を軽減することに寄与していると見なすことが出来る。

4 考察

今回、産学連携による PBL を実施したことによって、

表2　PBL 第三における工数比率

工程	要求定義	設計	実装	テスト	合計
工数比率(%)	38	19	17	26	100

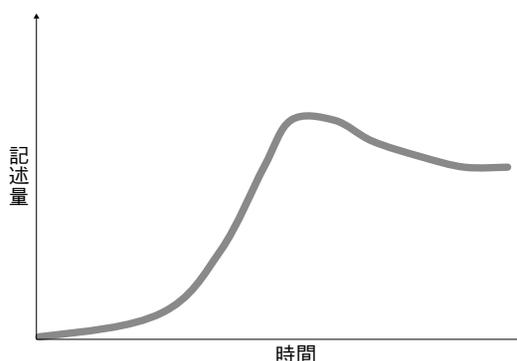



図6　形式的なシステム記述作成過程 [HALL1990]

形式手法の導入に関する有用な知見が得られた。実践的な教育としての産学連携による PBL の効果に関しても、種々の成果や知見が得られたが、それに関しては別の機会に述べることにして、ここではフォーマルメソッド導入に関して議論する。

今回、形式手法に関心のある企業が、産学連携による PBL を利用して学生チームに自社内の開発プロジェクトの一部を担当させることによって、形式手法の適用事例の経験を持つことが出来たとともに、導入に関する見通しを得ることが出来た。

形式手法の適用事例は、国内外にわたって数多く存在する [IPA/SEC-2]。しかしながら、それらの適用事例を参考にして、直ちに自社の開発プロジェクトに適用することは容易ではない。形式手法の特質を理解し、かつ、形式手法適用の目的を明確に持っていなければ、個々の具体的な開発プロジェクトに効果的に形式手法を適用することは難しいであろう。

今回の PBL 第二では、顧客としての企業も開発担当の学生チームも、前述のように形式手法自体に対する知識と経験を有しておらず、そのため、具体のソフトウェア開発のどこにどのように適用すれば良いかも分からぬままに、開始した。その結果、学生チームは、少なからず苦労を強いられたわけであるが、その学生チームの貴重な経験や知見を企業内に残すことに成功した。その理由は、他の適用事例と違って、企業側が今回のプロジェクトを自らの問題として認識出来たからであると考える。PBL のテーマとして自社内の具体的な開発プロジェクトの一部を提供したことと、PBL 担当窓口としての現役技術者を充てたことによって、社内の問題及び活動として、日常の開発業務の中での形式手法の適用の一方法が見えたのではないかと思われる。形式手法の初学者としての学生の経験は、社内に形式手法を導入する際の形式手法習得の過程と適用を事前に擬似経験したことに相当して、今後の導入に対する見通しを得ることに役立った。

図6に A. Hall が示した形式仕様記述のプロセスを示す [HALL1990] 。生産される記述物の量が時間とともにどのように変化するかを表している。ここで、特徴的なことは、立上がりの遅さである。しかし、開発対象を十分に理解して初めて明解な仕様を厳密に記述出来るとい

う形式手法の特質を思えば、それは当然のことである。

PBL 第二では、顧客役の企業も開発担当の学生チームも、このことは話には聞いていたかも知れないが、体験はしていないので、プロジェクトがどのように進展していくのかに関する見通しを持つことが出来なかった。プロジェクト初期の段階で、対象ドメインの理解並びに形式手法の習得やその適用方法の提案に関して、時間ばかりが経過してなかなか進捗していないかのように見えていたときに、スーパバイザ役の教員としては、我慢強く見守ることに徹した。

PBL 第二での経験に基づいて、PBL 第三では新たな開発対象に取り組んだにも拘わらず、表2 に示したように要求定義フェーズで時間をかけてドメインの理解と仕様記述を行い、後工程での効率的な開発に続けることが出来た。

5 おわりに

本稿では、産学連携による PBL における形式手法の適用事例について教員の立場から概要を報告した。企業における実際の開発プロジェクトを題材とすること、現役の技術者が学生の指導にあたることなどの貴重な機会を得て、学生は多くのことを学ぶことが出来た。筆者としても、形式手法を企業の開発現場に導入する際の１つの方法として産学連携による PBL が有効であることが実証出来たことの意義は大きい。PBL による形式手法の導入及び適用の事例研究として、更なる具体的な分析と考察を行う予定である。

併せて、企業の立場からの形式手法の適用事例に関する分析と評価を取りまとめて公表することも検討している。現在は、より密な産学連携の形態で、2011 年後期と 2012 年前期の PBL を実施中である。これにより、新たな展開とそれによる成果が期待出来るが、実施後にその報告を行いたい。

謝辞

本稿で紹介した PBL に参加した九州大学大学院システム情報科学府情報知能工学専攻社会情報システム工学コースの学生諸君、並びに、株式会社富士通九州ネットワークテクノロジーズの皆様に感謝します。

参考文献

[ARAKI2011] 荒木 , 日下部 , 大森 , 岩本 , 篠沢 , 本田 , 宮下 , 山田 , 岩崎 , 井上：産学連携によるフォーマルメソッド導入事例 ― 仕様の品質向上を目指して ―, ソフトウェア・シンポジウム 2011, 長崎 , 2011 年 6 月

[HALL1990] J. A. Hall：Seven Myths of Formal Methods, IEEE Software, Vol.7, No.5, pp.11-19, 1990

[IPA/SEC-1] IPA/SEC：ソフトウェア開発データ白書 2010 - 2011, 2010 年 11 月

[IPA/SEC-2]「形式手法適用調査」報告書 , 2010 年 7 月 , http://sec.ipa.go.jp/reports/20100729.html

[SHINOZAWA2011] 篠沢 , 他：高品質な実システムの開発における形式手法の適用 , 情報処理学会九州支部若手の会セミナー 2011 講演論文集 , pp.9-12, 鹿児島 , 2011 年 9 月

[YAMADA2011] Shinya Yamada, et al.: An Introduction of a Formal Method in PBL：A Case Report, Joint Workshop on Software Science and Engineering, Seoul, June 2011

一般社団法人TERASの紹介（前編）

安心･安全･快適な社会のために全ドキュメントのトレーサビリティを目指す

キャッツ株式会社
マネージャ
TERAS 広報委員
穴田　啓樹

キャッツ株式会社
副社長
TERAS 理事
渡辺　政彦

名古屋大学
教授
TERAS 技術委員会委員長
高田　広章

TERAS

URL：http://www.teras.or.jp/

近年、機能安全規格 IEC 61508、ISO 26262に代表されるように、安全に対する説明責任が課されている。今までは「正しい製品」を作ることが安全性を証明する手段だったが、システムが極めて複雑化しており、完璧な製品を作ることは難しくなっている。今後は製品に問題があった場合にも同様に、企業として十分な努力をしたか、言い換えると「正しい方法」で作ったかを説明･証明することが求められ、第三者による検証や、より精緻な品質の監査が必要となる。このソフトウェア品質監査で最も重要になるのがトレーサビリティである。本稿では、トレーサビリティを実現するためのオープンな「ツールプラットフォーム」を提供する「TERAS※1」について、本号と29号（予定）の2回にわたり解説する。今回はトレーサビリティの概要についてふれ、次回はその詳細と実証評価の状況を説明する。

1　TERAS 発足の背景

組込みシステム産業の市場規模は国内総生産の 10% 超に相当している。また、輸出製品に占める組込みソフトウェア搭載製品の割合は 50% を超えるなど、組込みソフトウェアは日本の輸出産業を支える製品付加価値の源泉と言える。その一方で、組込みソフトウェアが原因で製品・システム・サービスに不具合が発生し、企業に社会的責任が問われるケースが増えている（図 1）。2010 年の自動車の米国リコール問題を契機に、組込みソフトウェアの安全性を向上させるとともに、安全性に関する説明責任を果たすことの重要性がますます高まってきている。日本の産業の競争力の源泉とも言える組込みソフトウェアにかかわる信頼性・安全性の確保は、極めて重要な課題であり、早急な対策が求められる。

脚注

※1　TERAS：Tool Environment for Reliable and Accountable Software

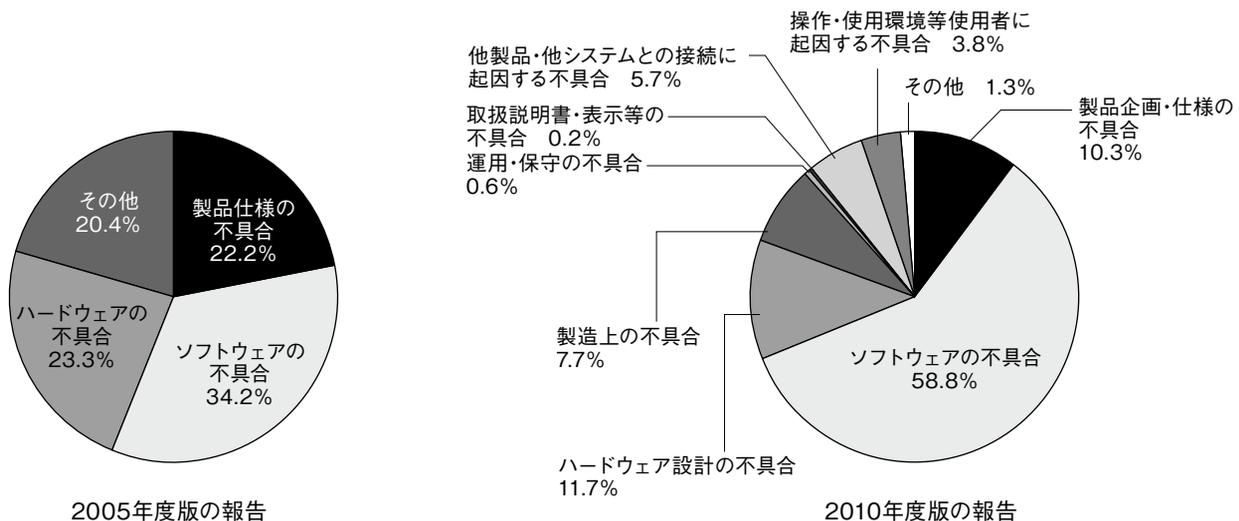



図1　不具合の原因の割合[経済産業省HP-1]

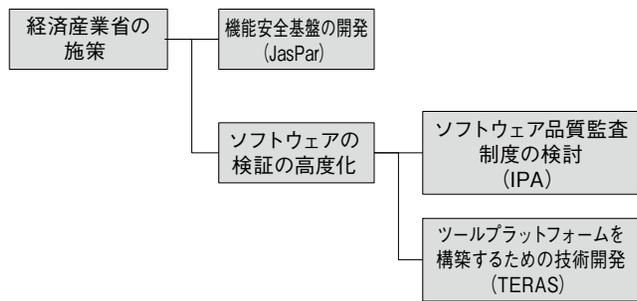



図2　経済産業省の施策[経済産業省HP-2]

また、近年、IEC（国際電気標準会議）で、製品分野毎にシステムに対する安全面の要求事項をまとめた規格が次々と制定されている。このような状況を受けて経済産業省では組込みシステム基盤開発事業として以下の2つの対応策をとっている（図2）。

第一に機能安全基盤の開発である。日本の基幹産業であり高度な制御が求められる自動車分野に着目して、2010年から3カ年で機能安全規格に対応した解説書の策定、国際的に安全と認められたソフトウェア開発手法の具体化、機能安全に対応した基盤ソフトウェアの開発が実施される計画である。ここで得られた成果はロボットなどの他産業にも横展開される予定である。

第二にソフトウェアの検証の高度化である。従来の人海戦術による検証ではその妥当性を評価するのに限界があり、開発者等と利害関係の無い第三者が製品やシステムの信頼性・安全性等を検証する枠組の検討がなされている。第三者が検証を実施、その妥当性を検証するには、その製品等の要件から設計・実装・検証・変更にいたるまで、正しい方法で作られたことを確認するために、開発成果物のトレーサビリティが必要である。このトレースを人手で間違いなく行うのは困難であり検証用ツールが必要となるが、すべての成果物のトレーサビリティを扱えるツールは存在していない。そこで、2011年から3カ年でオープンなツールプラットフォームを構築する事業が計画され、TERASが発足した。

2　他分野での状況

「トレーサビリティ」という言葉は幅広く使われており「追跡出来ること」「追跡可能性」を指している。身近なところでは、食品のトレーサビリティについて耳にする機会が多い。本原稿の執筆時点(2011年11月)では、牛肉、米・米加工品のトレーサビリティについての法律が施行されている。もともと食品分野では、HACCPやISO 9001などにより食品の安全や品質の確保が取り組まれてきた。しかし、狂牛病の発生や食品の産地偽装表示事件により食品や業界に対する不安が高まり、直接に影響の及ばない商品や産地の商品まで消費が落ち込む事態となった。そこで、生産・加工及び流通の履歴を確認出来ることが望まれた。2004年12月1日から牛肉のトレーサビリティが義務付けられ、2011年7月1日から米のトレーサビリティが義務付けられた。食品のトレーサビリティ法では、生産から販売・提供までの各段階を通じ、取引等の記録を作成・保存する。食品に問題が発生した際には流通ルートを速やかに特定し、その影響範囲を特定して対応する。このようにトレーサビリティをとることは、消費者の安全を守り、関連産業の発展を図ることを目的としており、組込みソフトウェア産業の状況と一致する。

3　トレーサビリティのあるべき姿

機能安全規格で求められていることの多くは、実は日本企業で既に実施されているとされる。しかし、影響度解析、すなわち、上流工程の変更における下流工程での影響についての検証については不十分と聞く。特に、欧米と比較すると、上流工程の設計コンセプトや設計の過程をたどることが出来ないケースが多く、これでは第三

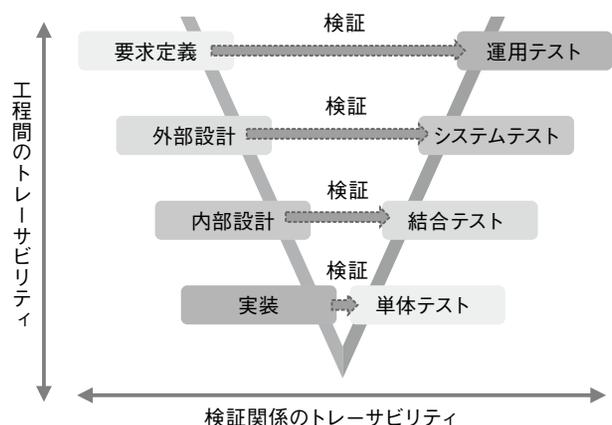



図3　2つの基本トレーサビリティ

者による検証は困難である。

それでは、何のトレーサビリティをとればよいのか。トレーサビリティは何かと何かをひも付けるものであり、組込みソフトウェア開発では主に２つの関係が基本となる。

第一に要求・設計・実装などの工程間の関係である。V字開発モデルでいえば、図３のように左上から右下に下がる工程間の関係で、上流の成果物と下流の成果物を結び付けるトレーサビリティである。例えば、この設計はどういう要求があってこの設計になったのか、この要求はどういう設計で実現されたのかを追跡するために利用する。

第二に要求・設計・実装と検証の間の関係である。V字開発モデルでいえば、左の工程と右の工程間の関係を結び付けるトレーサビリティである。例えば、この要求が達成されたのかを確認するのにどのように検証したのか、この設計が実装され、きちんと作られたことをどのように検証したのかを追跡するために利用する。機能安全規格に対応するにはこれらのトレーサビリティを利用して、設計変更時の影響分析や設計項目の実装・検証に漏れがないこと（カバレッジ）を証明する。

4 日本の強みを生かすトレーサビリティ

製品アーキテクチャには、組み合わせ型と擦り合わせ型がある（図４）。欧米企業は組み合わせ型製品が、日本企業は擦り合わせ型製品が得意とされている。

擦り合わせ型製品の開発では、機能要素、または品質特性と構造要素・部品が多対多に対応し、部品間の関連が強く、部品設計の擦り合わせが必要となる。これをタスク間の情報の相互依存度を示すDSM[※2]で確認してみる。図５のように、タスクがＡからＦに流れるとき、組み合わせ型製品と擦り合わせ型製品を比較すると、擦り合わせ型製品の開発では右上半分に×印が多くなり、各構造要素・部品間の設計が密接に関連していることが明らかになる。

擦り合わせ型製品の開発の他の特徴としては、要求の改善が挙げられる。欧米の製品開発では要求から設計・実装へとトップダウンに進むことが多いが、日本の製品開発ではトップダウンに進むとは限らず、下流工程から上流工程へ改善要求を出し、そこで擦り合わせが発生してより品質の高いものを生み出そうとする。実際に、改良・改善は日本のものづくりの強さの一因であり、今後の成長のためには、継続的な改良・改善を支援するプロセスが求められる。

ここで重要なのは、改良・改善は変更を意味することである。変更するにあたり、各構造要素の設計間の関係を追跡したり、要件がどこから出てきてどのように変化したか追跡したり、未定の要件を追跡したりしたくなる。このとき、前述のようにドキュメントのトレーサビリ

脚注

※2　DSM：Design Structured Matrix

図4　2種類の製品アーキテクチャ[藤本2004]

	A	B	C	D	E	F
A			×			
B	×			×	×	
C		×		×		
D		×				×
E	×	×	×	×		
F				×	×	

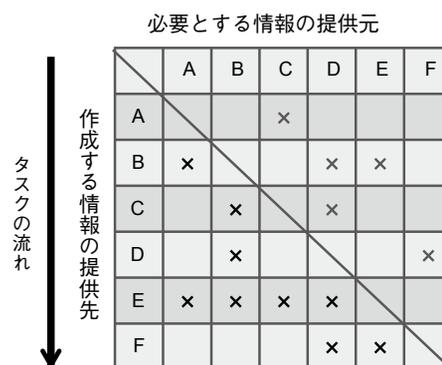



図5　擦り合わせ型製品の情報の相互依存度[TERAS HP-1]

ティがとられていなかったり、設計の過程が残されていなかったりすると変更の影響分析が出来ず、変更しにくいシステムになってしまう。その結果、徐々に改良・改善のフィードバックがかけられなくなる。つまり、日本の強みを生かすにはトレーサビリティが必要なのである。

5 TERASの事業活動

このように、日本の強みを生かすためにもトレーサビリティが必要であり、日本のものづくりを強くするツールプラットフォームを実現・普及していくことがTERASのミッションである。TERASは経済産業省の補助金事業と民間企業7社の拠出による民間事業で構成し、2011年から3カ年の開発フェーズでツールプラットフォームを構築・実証評価・ソフトウェア品質監査の啓蒙活動を推進していく（図6）。

ツールプラットフォームの開発では、2012年度に第一弾の製品リリースを目指して活動している。まず、日本企業の「擦り合わせ型開発プロセス」を調査・分析して「擦り合わせる」際に必要なトレーサビリティの格納方法を定義する。次に、そのデータ構造を扱うツールプラットフォーム仕様を定義して、プラットフォーム上でツールのプラグインが動作する環境を構築する（図7）。

図6　TERASの事業計画[TERAS HP-2]

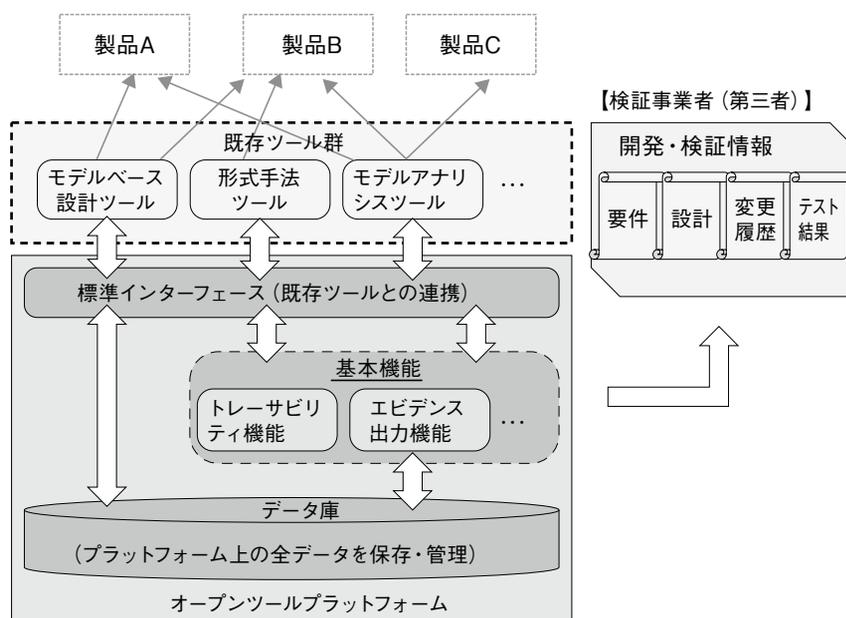



図7　オープンツールプラットフォームの概念[TERAS HP-3]

写真1　TERAS設立発表会の様子

この際、トレーサビリティのエンジンの基本部分は無償で公開し、ツールベンダーの参画を募る。また、ソフトウェア設計にかかわるすべての種類のドキュメントでトレーサビリティがとれる必要があるので、開発仕様を公開してオープン化活動を推進していく。

実証評価・啓蒙活動では、ツールへの改善要求を洗い出すために製品リリースに先駆けて実証評価会員を募り、トレーサビリティツールの導入を支援する。実証評価と並行して啓蒙活動事業では、定期的にセミナーやワークショップを開催する。ソフトウェア品質監査の考え方や、必要な取り組みについては、IPA/SECと協調しながら普及活動を推進していく。TERASとして開催した第一弾のイベントであるTERAS設立発表会（写真1：2011年10月5日、東京コンファレンスセンター・品川、5F 大ホール）では定員を超える312名の参加者が集まり、交流を深めた。来場者の業種は製造業関連が6割を超え、来場者アンケートではTERASへの高い関心が寄せられた。

6 おわりに

TERASでは実証評価に協力していただける企業を募集している。関心を持たれた企業はぜひ、TERAS事務局へ連絡されたい。次回の寄稿では、TERASの技術の詳細・実証評価の分析結果について解説する。

問い合わせ先

・TERAS事務局

神奈川県横浜市港北区新横浜2-11-5 川浅ビル

電話：045-473-2191

E-mail：secretariat@mail.teras.or.jp

URL：http://www.teras.or.jp/

参考文献

[TERAS HP-1]【講演3】設計トレーサビリティを支援するツールプラットフォームの概要, http://www.teras.or.jp/

[TERAS HP-2]【講演2】TERASが目指す安心・安全・心地よい社会, http://www.teras.or.jp/

[TERAS HP-3]【講演1】ソフトウェア品質監査の重要性について, http://www.teras.or.jp/

[経済産業省 HP-1] 経済産業省：組込みソフトウェア産業実態調査報告書, http://www.meti.go.jp/policy/mono_info_service/joho/ESIR/index.html

[経済産業省 HP-2] 経済産業省：組込みシステム基盤開発事業, http://www.meti.go.jp/main/yosangaisan/2012/pr/pdf/sangi_04.pdf

[農林水産省 HP] 農林水産省：http://www.maff.go.jp/j/syouan/seisaku/trace/index.html, http://www.maff.go.jp/j/syouan/seisaku/trace/pdf/tebiki_rev.pdf, 食品トレーサビリティシステム導入の手引き（食品トレーサビリティガイドライン）

[藤本2004] 藤本隆宏：日本のもの造り哲学, 日本経済新聞出版社, 2004（一部修正）

CEA-LIST
フランス原子力･代替エネルギー庁システム統合技術応用研究所
ー安全性とセキュリティの高いシステムを目指してー

French Embassy in Tokyo
Japan Representative
ILJIC Thomas

CEA-LIST Institute
International Marketing
NAHHAL Karima

CEA-LIST Institute
Deputy Director
APOLINARSKI Xavier

URL：http://www-list.cea.fr/

CEA-LIST[※1]は、スマート･ディジタルシステムに注目した研究開発を実施する研究所であり、フランス原子力･代替エネルギー庁の1機関である。フランスのパリ郊外にあり、700名の研究者と技術者（PhD120名含）を擁し、〈情報通信技術（ICT）〉〈エネルギー〉〈交通〉〈セキュリティ&防衛〉〈健康〉〈製造〉分野に関する研究を行っている。

各分野固有の研究の成果の公表から技術移転まで、技術革新の全プロセスをカバーし、基礎研究から製品開発に至るまで、幅広い業務を網羅している。

特に、経済的、社会的ニーズの高い以下の3つの分野に力を入れている。

・インタラクティブシステム
　インタラクティブロボット工学、バーチャルリアリティ、感覚インターフェース、ビジョン&マルチメディア、通信システム
・組込みシステム
　マルチプロセッサ･アーキテクチャ、組込みシステムの安全性とセキュリティ、ソフトウェア&システムエンジニアリング
・センシングシステム
　革新的なセンサー、計測システム、データ分析、非破壊試験、電離放射線計測学

1 研究内容とその特色

CEA-LIST は、技術革新及び技術移転により産業界の競争力を高め、支援することを目的としている。パートナーである大企業や中小企業と、研究者との間には、長期にわたる研究開発の継続が可能であり、パートナーのニーズを十分に理解したうえで、活動を実施している。そして、産学官連携等、関係各機関との強力なネットワークにより高い研究レベルを維持し、多くの技術革新を可能としている。

また、CEA-LIST では将来の社会的ニーズに対応するため、環境問題を考慮し、ユーザフレンドリーなヒューマンマシンインターフェース、安全で信頼性の高いシステム、インテリジェントなエネルギー管理システム、産業用エコデザイン導入プロセス用のツールなどを開発している。

パートナーは100社に及び、1年に約200件の契約を締結し、2001年以降、ハイテク・ベンチャー企業11社を設立している。現在、全世界で237件の特許ポートフォリオ並びに74件のライセンスを所有している。

2 パートナーシップとエコシステムを中心としたビジネスモデル

CEA-LIST は、フランスのリーダー的 IT クラスターである DIGITEO、コンプレックス・システム[※2]を専門とする Systematic Paris-Region 他、競争力あるクラスター、Cap Digital、Aerospace Valley、Moveo などワールドクラスのエコシステムプロジェクトと連携のもと、

研究開発を進めている。

約100件の欧州プロジェクトに協力し、そのうちの12件でプロジェクト・リーダーを務め、また複数のネットワーク・オブ・エクセレンス（NOE）、プラットフォーム及びクラスター（組込みシステム用のArtemis、ロボットの向けのEurop、持続可能生産のManufuture、スマートグリッドなどに参画し）、特にセキュリティ分野では欧州圏の研究計画に大きく貢献している。システム設計から、工業化を目指すプロトタイプまで、イノベーション・チェーン全体を通じて、パートナーと一貫性のある共同技術のロードマップ（長期的、短期的）に基づき活動している。

技術的ビジョンは、主に、CEA-LISTと産業界のパートナーとの間に存在する30の共同研究室やR&Dプログラムにおいて作成されている。CEA-LISTのビジネスモデルは、国際レベルで開発を行うテクノロジープロバイダ（OEMサプライヤ、ソフトウェアの開発者）と協力し、エンドユーザのニーズを満たす技術の開発を行うことにある。

3 安全性とセキュリティ分野に注目

もともと、CEA-LISTは特に原子力、航空学の重要なアプリケーションの開発分野において、そのソフトウェアの専門知識を構築してきた。その間に、他のエンジニアリング分野や新技術分野におけるソフトウェアの開発比率が徐々に高まってきた。例えばロボット工学、輸送、ホームオートメーション、健康、エネルギーなどの分野である。ソフトウェアは、製品の付加価値の大部分を占め、以前は機械で実行された機能を変換（例：電子燃料噴射装置）、オペレーショナル・ハードウェアの進化に容易かつ迅速に適応出来るものであった。従って、エンジニアは、ますます複雑になるシステムまたその他の問題（市場投入までのタイムラグの短縮、機能のレベルアップ、安全性とセキュリティ問題の重要性）など、乗り越えなくてはならない問題が山積している。これらの新たな課題にアプローチするために、CEA-LISTは、一方ではソフトウェアをサポートするハードウェアという観点から、他方では開発プロセスの自動化という観点から、Rise in Abstraction（抽象概念を源とする）設計ツールの強化を目指している。

その設計を正しく行うため、CEA-LISTでは、最新技術や研究成果に基づき、システム・バリデーション及びチェックを行うツールの設計も手掛けている。これらのツールは、開発の初期から正しいモデリングやデザインを可能とするもので、その動作時間とコストを削減するなど、開発の全プロセスに適用することが出来るものである。

CEA-LISTでは、仕様からソースコードまで、開発プロセスの各段階で、安全性とセキュリティに関する要求を満たすよう考慮している。これらの相互領域における要求は、ますます大きくなっている。その目標の1つは、フォールト・トレランスやアンチウィルス対策などに適応する一般的な対策ソフトを設計するために必要な共通点を明らかにすることである。

これら技術は、モデリングの主要工業規格に基づき開発されている。そういった意味でCEA-LISTはOMG※3やAUTOSAR※4などの国際標準化機関でも、非常にアクティブなプレーヤとして活動している。例えば、MARTE国際標準※5（embedded real-time modelling language）の導入時にも、主要な役割を果たした。

Papyrusという Eclipse※6 プラットフォームのUML2ツールはCEA-LISTの成功例の1つである。

この基本モデリングツールを中心として、CEA-LISTでは、前述のコンプレックス・システムの開発にかかわる様々な課題を扱うために、コンパニオンツールセットの提供も行っている。

問い合わせ先
・在日フランス大使館
原子力・代替エネルギー庁　CEA最先端技術局　アタシェ
東京都港区南麻布4-11-44
電話：03-5798-6339
E-Mail：thomas.iljic@snaft.jp

脚注

※1　CEA-LIST：フランス原子力・代替エネルギー庁システム統合技術応用研究所, Commissariat a l'Energie Atomique et aux Energies Alternatives, Laboratoire d' Integration des Systemes et des Technologies
※2　コンプレックス・システム：IPA/SECの統合系プロジェクトで扱う統合システムの概念にほぼ同等。
※3　OMG：Object Management Group, http://www.omg.org/
※4　AUTOSAR：AUTomotive Open System ARchitecture
※5　MARTE国際標準：http://www.omgmarte.org/
※6　Eclipse：http://www.eclipse.org/papyrus/

Column

就職はなぜ難しいのか

IPA顧問 学校法人・専門学校HAL東京 校長
鶴保 征城（つるほ せいしろ）

SEC journal 24号で「はずさない就職活動とは」という記事を書かせていただいたが、その後、東日本大震災や景気低迷などの影響が就職戦線に更なる打撃を与えている。本稿では再度、就職問題の本質を考えてみたい。

未曾有とも言われる現在の就職難は、大震災や景気低迷という一過性の要因だけではなく、日本がここ20年以上も直面している構造的な問題が原因しているのではないかと思う。筆者はそれを以下の4項目に整理した。

①大学生及び大学の質の低下

24号でも書いたが、日本の若者人口は団塊ジュニアが成長した1993年にピークをつけた以降、右肩下がりに減少している。その一方で、大学進学率は25％程度（1993年）から2009年には50％を超えるまでになった。大学数も同様に、500校程度から今では800校を超えている。入学試験はというと、無試験ないしは2科目試験のように大学入試の簡略化が一般化してきた。

結果として、「漢字が読めない」「割り算が出来ない」など、大学生の質の低下が巷間、喧伝されている。いろんな大学の先生方に聞いてみると、定員割れを起こした年はほぼ全入であり学生のレベル低下が著しいようだ。本稿ではこれ以上詳しく述べないが、2人に1人が進学する以上、この問題は今後も避けられないと思う。

学生の質以上に深刻なのが、大学そのものの質の低下である。急増した大学では魅力あるカリキュラムが不足しているし、まして、下位レベルの学生を引っ張り上げて就職に漕ぎつけさせる熱意と技量のある教員がそんなに多くいるとは思えない。

②産業構造の変化と要求されるスキルの変質

就職問題で取り上げられることは少ないが、筆者はこれが最大の論点だと思っている。

戦後の高度成長期を通じて、農業や漁業の一次産業が衰退したことは周知の通りである。これに続いて、直近20年間に起こり、今一層拍車がかかっているのが二次産業の海外移転である。1990年から2010年の20年間で、一次・二次産業は564万人の雇用を失い、三次産業は543万人雇用を創りだした。

一次・二次産業と三次産業では、要求されるスキルが真逆といっていいほど異なる。前者では人との接触が苦手でも何とかなるが、後者では組織の一員として対人関係に気を遣いながら仕事を進める能力が求められる。リーマンショック後に派遣切りが多発した折に、多くの人が（雇用を申し出た）サービス業への転職を断ったことは記憶に新しい。

③オペレーションから戦略へ

二次産業の工場などが海外に移転するに伴って、国内に何も残らないのかというとそうではない。国内には、戦略、金融財務、情報システム、技術開発、先端工場（マザー工場）、海外拠点の統括、人材育成・教育などの重要な業務が残る。企業の仕事を戦略とオペレーションに大別すると、日本は、生産性と品質を合言葉に、ひたすらオペレーションに磨きをかけて勝ち抜いてきた。それらは今後も重要であるが、残念ながらオペレーションの実行そのものは海外勢に任せざるを得ない。日本勢に期待されるのは、戦略レベルの仕事ということになる。

④大企業への執着

もう一つのミスマッチは、企業規模に関するものである。日本における求人倍数（求人数と求職数の比率）は、2000年に0.99を記録した以外、常に1以上であり、直近のデータでも1.62となっている。総数として求人が不足していることはなく、依然として失業率は低い。ところが、これを従業員1,000人以上の企業と1,000人以下の企業でみると、前者が20年以上にわたって1を超えたことがないのに対して、後者は常に2～4倍、直近でも1.86もある。多くの学生が大企業中心の就職活動を行っているのではないかと思うが、中小企業が常時人手不足であることを頭の中に入れておくべきだと思う。

以上、日本の構造的な問題が、若者の就職を直撃していることを述べた。若者に相当の覚悟をしてもらうと同時に、彼らを取り巻く多くの関係者の適切なナビゲーションが必要だ。

改善の相乗効果が創発的なイノベーションに通じる

創発的破壊
未来をつくるイノベーション

米倉誠一郎　著

ISBN：978-4-903908-27-4
ミシマ社刊
四六判・292頁
定価1,785円（税込）
2011年6月刊

ソフトウェアにかかわる者は、自ずと社会にかかわっている立場である。この本は技術書ではないが、今の社会がどのように動こうとしているか、その認識を深めるには適切なガイドになろう。

ソフトウェアは、発注者の要求を出発点として作られることが多い。故に、しばしば発注者の立ち位置に影響される。ソフトウェア開発の受注者は、発注者の要望に応え、イノベーティブなソフトウェアを開発し、発注者のビジネスイノベーションに関与することは充分可能である。

この本は、イノベーションに関して今日的課題から例と共に説明している。この本において繰り返し出てくるものの一つは、日本人のグローバルな場への更なる登場であろう。これは、日本語という障壁により守られているとも言われるITビジネスにおいては、十分考えなければならない。

特に、冒頭に述べた背景を持つ私達ソフトウェア開発に関係する者には64頁にある次の文章が参考になる。

「既存技術で既存市場を深堀するのは日々のカイゼン・改良にすぐれた「経営管理者的企業家」である。アバナシーたちのもう一つのすぐれた視点は、それまでのイノベーション論で排除されてきた日々のカイゼン・改良あるいは地道なプロセス・イノベーションを立派なイノベーションと認定し、この「通常的革新」こそが競争力の源泉としたことであった。」

（新谷　勝利）

「コーディング規約」作成のすすめ

SEC BOOKS
組込みソフトウェア開発向け コーディング作法ガイド
[C++言語版]

IPA/SEC 編著

ISBN：978-4-274-50316-0
オーム社刊
B5変型判・192頁
定価1,800円（税込）
2010年7月刊

本書はタイトルが「コーディング作法ガイド」であるため、コーディング規約として利用出来る材料であることはあまり知られていない。行儀の良いコーディングの仕方を教えてくれるガイドブックには違いないが、本書を開いてみると、そのまま組込みソフトウェア開発のコーディング規約として使えることが分かる。

組込みソフトウェア開発において、要求定義やソフトウェア設計を避けてコーディング作業を行うことは出来ないが、コーディング規約を定める作業を省いても、コーディング作業は行える。新規にコーディング規約を作らなければならないと思うと面倒なため、つい非優先作業にしがちであるが、そのまま利用出来る材料があるのなら利用しない手は無い。ソフトウェアの品質向上のための費用対効果を考えるとむしろ優先すべきである。

本書は、近年の組込みソフトウェアの大規模化によって使われる「C++言語」に対応し、組込みソフトウェア開発共通に規約化した方が良いルール、プロジェクトの特性に従って選択すれば良いルール等が分かりやすく示されている。

コーディング規約を定めることが、組込みソフトウェアの品質確保の近道であることを疑わず、まず本書を開いてみることをお勧めする。

なお、本書に先だって発行された「C言語版」は、JIS X 0180「組込みソフトウェア向けコーディング規約の作成方法」として2011年4月にJIS制定されている。

（松田　充弘）

編集後記

激動の2011年が過ぎて、2012年を迎えました。振り返ってみると3月11日の東日本大震災に端を発して、日本が大きく変わろうとしています。

この歴史的な大転換のうねりの中で、今までの延長線上にはない新しい社会の構築を目指して“創発的イノベーション”（本号の書評参照）が期待されているようです。

昨年は、SEC journalに採録された論文の中から、3編の論文を優秀賞に選定し、10月のIPAフォーラムにて表彰しました。その選定の経過などは本号に掲載されています。ソフトウェアの開発現場や大学からの論文の成果を広く共有していただき、ソフトウェア・エンジニアリングの更なる発展のために、本年も多くの方からの積極的な論文投稿をお待ちしております。

SECは、“創発的イノベーション”を生み出す原動力として、我が国のソフトウェア・エンジニアリングの基盤強化に、より一層注力して、今まで以上に皆様のお役に立っていきたいと考えております。本年もどうぞよろしくお願い致します。

（久保）

編集部より

次世代のソフトウェア・エンジニアリング等に関して、忌憚のないご意見をお待ちしております。
FAX、または下記のメールアドレス宛にご連絡ください。
SECジャーナル編集部宛　e-mail：sec-journal_customer@ipa.go.jp

SEC journal 編集委員会

新春を迎える石畳の街　（撮影：金沢成恭）

SEC journal® 第7巻第4号（通巻29号）　2012年1月12日発行

編集兼発行人　〒113-6591　東京都文京区本駒込2-28-8　文京グリーンコート センターオフィス16階
独立行政法人情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター　所長　松田　晃一
Tel.03-5978-7543　Fax.03-5978-7517
http://sec.ipa.go.jp/

お知らせ

SEC journal 論文募集

IPA 技術本部 ソフトウェア・エンジニアリング・センターでは、下記の内容で論文を募集します。

応募様式は、下記のURLをご覧ください。
http://sec.ipa.go.jp/secjournal/papers.html

論文テーマ

ソフトウェア開発現場のソフトウェア・エンジニアリングをメインテーマとした実証論文

- 開発現場への適用を目的とした手法・技法の詳細化・具体化などの実用化研究の成果に関する論文
- 開発現場での手法・技法・ツールなどの様々な実践経験とそれに基づく分析・考察、それから得られる知見に関する論文
- 開発経験とそれに基づく現場実態の調査・分析に基づく解決すべき課題の整理と解決に向けたアプローチの提案に関する論文

論文の評価基準

a. 実用性(実フィールドでの実用性)
b. 可読性(記述の読みやすさ)
c. 有効性(適用した際の効果)
d. 信頼性(実データに基づく評価･考察の適切さ)
e. 利用性(適用技術が一般化されており参考になるか)
f. 募集テーマとの関係

応募要項

投稿締切り

年4回、3ヵ月毎に締切り、締切り後に到着した論文は自動的に次号審査に繰り越されます。
(応募締切:1月・4月･7月・11月各月末日)
締切り後、査読結果は1ヶ月後に通知
詳細スケジュールについては、投稿者に別途ご連絡いたします。

提出先

独立行政法人情報処理推進機構　技術本部　ソフトウェア・エンジニアリング・センター内　SEC journal事務局
eメール:sec-ronbun@ipa.go.jp

その他

- 論文の著作権は著者に帰属しますが、採択された論文については SEC journalへの採録、ホームページへの格納と再配布、論文審査会での資料配布における実施権を許諾いただきます。
- 提出いただいた論文は返却いたしません。

論 文 賞

SEC journalでは、毎年SEC journal論文賞を発表しております(候補論文が少ない場合は、翌年の審議とする場合有り)。受賞対象は、SEC journal掲載論文他投稿をいただいた論文です(論文賞は最優秀賞、優秀賞、SEC所長賞からなり、それぞれ副賞賞金100万円、50万円、20万円)。

論 文 分 野

品質向上・高品質化技術
レビュー・インスペクション手法
コーディング作法
テスト／検証技術
要求獲得・分析技術、ユーザビリティ技術
見積り手法、モデリング手法
定量化・エンピリカル手法
開発プロセス技術
プロジェクト・マネジメント技術
設計手法・設計言語
支援ツール・開発環境
技術者スキル標準
キャリア開発
技術者教育、人材育成

SEC journal バックナンバーのご案内

詳しくはhttp://sec.ipa.go.jp/secjournal/をご覧ください。

ESxR特集号

No.22

No.23

No.24

No.25

No.26